

**SISTEM KLASIFIKASI AKTIVITAS MANUSIA MENGGUNAKAN
SENSOR *ACCELEROMETER* DAN *GYROSCOPE* DENGAN
METODE *K-NEAREST NEIGHBOR* BERBASIS ARDUINO**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Fadhilatur Rahmah
NIM: 145150300111036



PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

SISTEM KLASIFIKASI AKTIVITAS MANUSIA MENGGUNAKAN SENSOR
ACCELEROMETER DAN GYROSCOPE DENGAN METODE K-NEAREST NEIGHBOR
BERBASIS ARDUINO


SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik


Disusun Oleh :
Fadhilatur Rahmah
NIM: 145150300111036

Skripsi ini telah diuji dan dinyatakan lulus pada
1 Agustus 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Hurriyatul Fitriyah, S.T, M.Sc
NIP: 19851001 201504 2 003

Dosen Pembimbing II


Issa Arwani, S.Kom, M.Sc
NIP: 19830922 201212 1 003

Mengetahui
Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2 Agustus 2018



Fadhilatur Rahmah

KATA PENGANTAR

Puja dan puji syukur kami panjatkan kehadiran Allah SWT yang mana atas limpahan rahmat, taufik serta hidayah-Nya, skripsi ini dapat terselesaikan. Skripsi yang berjudul **“Sistem Klasifikasi Aktivitas Manusia Menggunakan Sensor Accelerometer Dan Gyroscope Dengan Metode K-Nearest Neighbor Berbasis Arduino”** ini disusun untuk memenuhi sebagian persyaratan dalam memperoleh gelar Sarjana Teknik.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada pihak-pihak yang telah bersedia untuk memberikan bantuan dan dukungan demi kelancaran penyusunan skripsi ini diantaranya:

1. Ibu Hurriyatul Fitriyah, S.T, M.Sc dan bapak Issa Arwani, S.Kom, M.Sc selaku dosen pembimbing skripsi yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng selaku ketua Program Studi Teknik Komputer Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Segenap dosen dan karyawan Fakultas Ilmu Komputer pada umumnya dan dosen Teknik Komputer Universitas Brawijaya pada khususnya, atas ilmu dan nasehat yang diberikan selama ini.
5. Orang tua dan saudara penulis (Ikhlashul Amaliyah dan Abdurahman Nafis), seluruh keluarga besar penulis dan keluarga Bapak H. Fathoni yang selalu memberikan doa, motivasi, kasih sayang serta dukungan moril dan materil sebagai penyemangat dalam menyelesaikan skripsi ini.
6. Teman-teman satu program studi yang telah membantu dalam menyelesaikan penelitian, Musada Teguh, Rizka Ayudya, Nafisa, Ingrid, Shelsa, Cindy, Syahriel, Naufal, Mas Zainul, Ayu Dewi dan seluruh keluarga besar Teknik Komputer seluruh angkatan khususnya angkatan 2014.
7. Pengurus Yayasan Karya Salemba Empat dan Para Donatur yang telah membantu serta seluruh teman-teman Paguyuban Karya Salemba Empat Universitas Brawijaya.
8. Pengasuh PPSP Al-Ishlahiyyah Bu Nyai Hj. Ummi Machmudah, teman-teman tercinta Mbak Zumi, Aini, Nahariya, Nurul, Jannah, Alfi, Lina, Persma, Jihan, Erna, Iemha, Naila dan teman-teman satu pesantren lainnya.
9. Teman-teman organisasi Korps Sukarela Universitas Brawijaya (KSR UB) dan teman-teman POROS FILKOM UB yang telah memberikan dukungan.
10. Seluruh pihak yang telah membantu yang tidak bisa disebutkan satu persatu.

Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Malang, 2 Agustus 2018

Penulis

frrahmah@gmail.com



ABSTRAK

Teknologi pengenalan aktivitas manusia memungkinkan sebuah sistem mendeteksi aktivitas sederhana yang dilakukan manusia, seperti berdiri, duduk, berbaring, berjalan, berlari dan lain-lain menggunakan kamera atau sensor. Sistem pengenalan aktivitas manusia berbasis kamera memiliki kekurangan tidak adaptif terhadap cahaya sehingga akurasi yang didapatkan kurang baik, sedangkan sistem berbasis wearable sensor yang menggunakan banyak sensor menimbulkan ketidaknyamanan saat digunakan dan masalah daya tahan baterai. Sehingga pada penelitian ini dibuat sistem yang dapat mengklasifikasi aktivitas sederhana yang dilakukan manusia menggunakan sensor MPU6050 yang sudah terdapat sensor *accelerometer* dan *gyroscope* serta menggunakan metode klasifikasi *k-Nearest Neighbor*. Input dari sistem ini adalah nilai pembacaan sensor *accelerometer* dan *gyroscope* yang dikirim menggunakan modul komunikasi *wireless* NRF24L01 kepada Arduino Mega sebagai perangkat yang melakukan klasifikasi dan menampilkan hasil klasifikasi pada *Serial Monitor* Arduino IDE. Pada penelitian ini terdapat sistem yang menggunakan satu sensor yang diletakan pada bagian tubuh yang ditentukan dan sistem yang menggunakan dua sensor. Metode klasifikasi *k-Nearest Neighbor* digunakan karena salah satu kelebihan metode ini yaitu tangguh terhadap *training* data yang *noisy* sesuai dengan karakteristik hasil keluaran sensor MPU6050 yang memiliki banyak *noise*. Dari hasil pengujian yang telah dilakukan masing-masing sebanyak 80 kali, didapatkan hasil akurasi tertinggi sebesar 93,75% untuk sistem yang menggunakan satu sensor dengan peletakan sensor pada paha dan sebesar 96,25% untuk sistem yang menggunakan dua sensor dengan peletakan sensor pada paha dan pinggang. Untuk pengujian waktu komputasi metode *k-Nearest Neighbor* dalam melakukan klasifikasi aktivitas manusia yang dilakukan masing-masing sebanyak 20 kali, didapatkan rata-rata waktu yang dibutuhkan sebesar 173,6 *milisecond* untuk klasifikasi yang menggunakan satu sensor dan sebesar 353,2 *milisecond* untuk klasifikasi yang menggunakan dua sensor.

Kata kunci: *accelerometer*, *gyroscope*, klasifikasi aktivitas manusia, *k-Nearest Neighbor*

ABSTRACT

Human activity recognition technology allows a system to detect simple activities by humans, such as standing, sitting, lying, walking, running and others using a camera or sensor. The camera-based human activity recognition system has a lack of adaptability to light so that the accuracy obtained is not good, while wearable sensor-based systems that use multiple sensors cause discomfort when used and battery life problems. So that in this study a system can be made that can classify simple activities by humans using the MPU6050 sensor which has an accelerometer and gyroscope sensor and uses the k-Nearest Neighbor classification method. Input from this system is the value of the accelerometer and gyroscope sensor readings sent using the NRF24L01 wireless communication module to Arduino Mega as a device that classifies and displays the classification results in Serial Monitor Arduino IDE. In this study there is a system that uses one sensor that is placed on the specified body part and a system that uses two sensors. The k-Nearest Neighbor classification method is used because one of the advantages of this method is that it is tough on noisy data training in accordance with the MPU6050 sensor output characteristics that have a lot of noise. From the results of the tests that have been conducted respectively 80 times, the highest accuracy results are 93.75% for systems that use one sensor with a sensor placement on the thigh and 96.25% for a system that uses two sensors with sensor placement on the thigh. and waist. For testing the computation time of the k-Nearest Neighbor method in classifying human activities 20 times each, obtained an average time required of 173.6 milliseconds for classification using one sensor and 353.2 milliseconds for classification using two sensors.

Keyword: *accelerometer, gyroscope, human activity recognition, k-Nearest Neighbor*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan masalah	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.1.1 Human Activity Recognition Using Wearable Accelerometer Sensor.....	5
2.1.2 Rancang Bangun Sistem Monitoring Deteksi Jatuh untuk Manula dengan Menggunakan Accelerometer.....	5
2.1.3 Identifikasi Aktivitas Manusia di Dalam Ruangan Menggunakan IP Camera dengan Metode Template Matching dan Algoritma Klasifikasi	6
2.2 Dasar Teori	7
2.2.1 Aktivitas Manusia	7
2.2.2 Metode <i>k-Nearest Neighbor</i>	7
2.2.3 Sensor Accelerometer	8
2.2.4 Sensor <i>Gyroscope</i>	9
2.2.5 MPU-6050	9
2.2.6 Arduino Nano	10
2.2.7 Arduino MEGA 2560.....	12

2.2.8 NRF24L01	14
BAB 3 METODOLOGI	15
3.1 Metode Penelitian	15
3.2 Studi Literatur	16
3.3 Analisis Kebutuhan	16
3.4 Perancangan	17
3.4.1 Perancangan Perangkat Keras	17
3.4.2 Perancangan Perangkat Lunak.....	18
3.4.3 Perancangan Pengambilan Data	18
3.5 Implementasi	18
3.5.1 Implementasi Perangkat Keras	19
3.5.2 Implementasi Perangkat Lunak.....	19
3.5.3 Implementasi Penentuan Posisi Peletakan Sensor.....	19
3.5.4 Implementasi Pengambilan Data Latih.....	19
3.6 Pengujian dan Analisis	20
3.6.1 Pengujian Fungsional Sistem.....	20
3.6.2 Pengujian Nilai K.....	20
3.6.3 Pengujian Akurasi Sistem Klasifikasi Aktivitas Manusia.....	20
3.6.4 Pengujian Waktu Komputasi Klasifikasi k- <i>Nearest Neighbor</i>	21
3.7 Kesimpulan.....	21
BAB 4 REKAYASA KEBUTUHAN.....	22
4.1 Gambaran Umum Sistem.....	22
4.1.1 Lingkungan Operasional Sistem	23
4.1.2 Batasan Desain Sistem	24
4.1.3 Asumsi Ketergantungan	24
4.2 Analisis Kebutuhan	24
4.2.1 Kebutuhan Fungsional.....	25
4.2.2 Kebutuhan Non Fungsional.....	25
BAB 5 PERANCANGAN DAN IMPLEMENTASI	27
5.1 Perancangan Sistem.....	27
5.1.1 Perancangan Perangkat Keras	27
5.1.2 Perancangan Perangkat Lunak.....	30

5.2 Implementasi Sistem	37
5.2.1 Implementasi Perangkat Keras	37
5.2.2 Implementasi Perangkat Lunak.....	38
5.2.3 Implementasi Penentuan Posisi Peletakan Sensor	49
5.2.4 Implementasi Pengambilan Data Latih	51
BAB 6 PENGUJIAN DAN ANALISIS.....	53
6.1 Pengujian Fungsional Sistem	53
6.1.1 Pengujian Pembacaan Nilai Sensor MPU6050.....	53
6.1.2 Pengujian Pengiriman Data Sensor	54
6.1.3 Pengujian Tampilan pada <i>Serial Monitor</i> Arduino IDE.....	56
6.2 Pengujian Nilai K	57
6.2.1 Tujuan Pengujian.....	57
6.2.2 Prosedur Pengujian	57
6.2.3 Hasil dan Analisis Pengujian	58
6.3 Pengujian Akurasi Sistem Klasifikasi Aktivitas Manusia	59
6.3.1 Tujuan Pengujian.....	59
6.3.2 Prosedur Pengujian	59
6.3.3 Hasil dan Analisis Pengujian	60
6.4 Pengujian Waktu Komputasi Klasifikasi <i>k-Nearest Neighbor</i>	61
6.4.1 Tujuan Pengujian.....	61
6.4.2 Prosedur Pengujian	61
6.4.3 Hasil dan Analisis Pengujian	62
BAB 7 PENUTUP	65
7.1 Kesimpulan.....	65
7.2 Saran	66
DAFTAR PUSTAKA.....	67
LAMPIRAN A DATA LATIH.....	69
A.1 DATA LATIH SISTEM DENGAN SATU SENSOR	69
A.1.1 PELETAKAN SENSOR DI PAHA	69
A.1.2 PELETAKAN SENSOR DI KAKI BAGIAN BAWAH	70
A.1.3. PELETAKAN SENSOR DI LENGAN	71
A.1.4 PELETAKAN SENSOR DI PINGGANG	72

A.1.5 PELETAKAN SENSOR DI KEPALA	73
A.2 DATA LATIH SISTEM DENGAN DUA SENSOR	74
A.2.1 PELETAKAN SENSOR DI PAHA DAN KAKI BAGIAN BAWAH	74
A.1.2 PELETAKAN SENSOR DI PAHA DAN LENGAN	76
A.1.3 PELETAKAN SENSOR DI PAHA DAN PINGGANG	78
A.1.4 PELETAKAN SENSOR DI PAHA DAN KEPALA.....	80
A.1.5 PELETAKAN SENSOR DI KAKI BAGIAN BAWAH DAN LENGAN	82
A.1.6 PELETAKAN SENSOR DI KAKI BAGIAN BAWAH DAN PINGGANG	84
A.1.7 PELETAKAN SENSOR DI KAKI BAGIAN BAWAH DAN KEPALA.....	86
A.1.8 PELETAKAN SENSOR DI LENGAN DAN PINGGANG	88
A.1.9 PELETAKAN SENSOR DI LENGAN DAN KEPALA	90
A.1.10 PELETAKAN SENSOR DI PINGGANG DAN KEPALA.....	92
LAMPIRAN B DATA UJI	74
B.1 DATA UJI SISTEM DENGAN SATU SENSOR.....	74
B.1.1 PELETAKAN SENSOR DI PAHA	74
B.1.2 PELETAKAN SENSOR DI KAKI BAGIAN BAWAH	76
B.1.3 PELETAKAN SENSOR DI LENGAN	78
B.1.4 PELETAKAN SENSOR DI PINGGANG	80
B.1.5 PELETAKAN SENSOR DI KEPALA	82
LAMPIRAN C KODE PROGRAM	84
C.1 KODE PROGRAM AKUISISI DATA SENSOR.....	84
C.2 KODE PROGRAM SISTEM PENERIMA DAN KLASIFIKASI DATA SISTEM DENGAN SATU SENSOR	85
C.3 KODE PROGRAM SISTEM PENERIMA DAN KLASIFIKASI DATA SISTEM DENGAN DUA SENSOR.....	90

DAFTAR GAMBAR

Gambar 2.1 Skema Posisi Sensor	5
Gambar 2.2 Pola Objek Saat Berdiri.....	6
Gambar 2.3 Modul Sensor MPU-6050	9
Gambar 2.4 Layout board Arduino Nano.....	10
Gambar 2.5 Arduino Uno	12
Gambar 2.6 Modul Wireless NRF24L01	14
Gambar 3.1 Diagram Alir Metode Penelitian.....	15
Gambar 4.1 Diagram Blok Sistem dengan Satu Sensor	22
Gambar 4.2 Blok Diagram Sistem dengan 2 Sensor.....	23
Gambar 5.1 Rangkaian Skematik Akuisisi Data.....	27
Gambar 5.2 Gambar Skematik Rangkaian Penerima Data	29
Gambar 5.3 <i>Flowchart</i> Pengambilan Data	30
Gambar 5.4 <i>Flowchart</i> Penerima Data.....	32
Gambar 5.5 <i>Flowchart</i> Fungsi k-NN().....	33
Gambar 5.6 Implementasi Perangkat Keras dengan Satu Sensor	37
Gambar 5.7 Implementasi Perangkat Keras dengan 2 Sensor.....	38
Gambar 5.8 Posisi Peletakan Sensor pada Bagian Tubuh.....	50
Gambar 6.1 Hasil Pengujian Pembacaan Nilai Sensor	54
Gambar 6.2 Hasil Pengujian Pengiriman Data Sensor	55
Gambar 6.3 Akurasi Hasil Pengujian Nilai k	58
Gambar 6.4 Hasil Akurasi Sistem dengan Satu Sensor	60
Gambar 6.5 Hasil Akurasi Sistem dengan 2 Sensor	61
Gambar 6.6 Hasil Pengujian Waktu Komputasi dengan Satu Sensor	63
Gambar 6.7 Hasil Pengujian Waktu Komputasi dengan 2 Sensor	64

DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino Nano 3.x	11
Tabel 2.2 Spesifikasi Arduino Mega 2650	13
Tabel 5.1 Keterangan Pin NRF24L01 dengan Arduino Nano	28
Tabel 5.2 Keterangan Pin MPU6050 dengan Arduino Nano.....	28
Tabel 5.3 Keterangan Pin Arduino Nano dengan Baterai	29
Tabel 5.4 Keterangan Pin NRF24L01 dengan Arduino Mega	29
Tabel 5.5 Hasil Perhitungan <i>Euclidean Distance</i>	34
Tabel 5.6 Hasil <i>Sorting</i> Nilai <i>Euclidean</i>	35
Tabel 5.7 Hasil Pengambilan 3 Data Pertama	36
Tabel 5.8 Kode Program <i>Import Library</i>	38
Tabel 5.9 Kode Program Inisialisasi Variabel	39
Tabel 5.10 Kode Program Fungsi <i>Setup()</i>	39
Tabel 5.11 Kode Program Pembacaan Nilai Sensor	39
Tabel 5.12 Kode Program Pengiriman Data	40
Tabel 5.13 Kode Program <i>Import Library</i> Penerima Data.....	40
Tabel 5.14 Kode Program Inisialisasi Variabel Sistem dengan Satu Sensor.....	41
Tabel 5.15 Kode Program Inisialisasi Variabel Sistem dengan 2 Sensor.....	41
Tabel 5.16 Kode Program Fungsi <i>Setup()</i>	41
Tabel 5.17 Kode Program Utama Penerima Data Sistem dengan Satu Sensor	41
Tabel 5.18 Kode Program Utama Penerima Data Sistem dengan 2 Sensor	42
Tabel 5.19 Kode Program Inisialisasi Kelas Klasifikasi dan Variabel	43
Tabel 5.20 Kode Program Pembacaan Data Latih dengan Satu Sensor.....	44
Tabel 5.21 Kode Program Pembacaan Data Latih dengan 2 Sensor	45
Tabel 5.22 Kode Program Perhitungan <i>Euclidean Distance</i> dengan Satu Sensor. 46	
Tabel 5.23 Kode Program Perhitungan <i>Euclidean Distance</i> dengan 2 Sensor.....	47
Tabel 5.24 Kode Program <i>Sorting</i> Data.....	47
Tabel 5.25 Kode Program <i>Voting</i> Jumlah Kelas yang Muncul.....	48
Tabel 5.26 Kode Program Penentuan Hasil Klasifikasi	49
Tabel 5.27 Posisi Peletakan Satu Sensor	50

Tabel 5.28 Kombinasi Posisi Peletakan 2 Sensor	51
Tabel 5.29 Kode Program Mengambil Data Latih	51
Tabel 6.1 Hasil Pengujian Tampilan pada <i>Serial Monitor</i> Arduino IDE	56
Tabel 6.17 Kode Program Fungsi <i>milis()</i>	62
Tabel 6.18 Hasil Pengujian Waktu Komputasi	62



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi yang semakin pesat menghasilkan berbagai inovasi teknologi dalam berbagai bidang. Salah satu bidang yang sedang ramai dikembangkan adalah *smart home* atau rumah pintar (Ivan, 2017). *Smart home* merupakan salah satu penerapan teknologi yang memungkinkan penghuni rumah dapat mengontrol berbagai peralatan rumah, terutama peralatan elektronik dengan satu genggaman, misal menggunakan smartphone atau perangkat yang lain (Ivan, 2017). Inti dari *smart home* adalah untuk memudahkan manusia melakukan pekerjaan rumah sehingga kualitas hidup semakin meningkat (Ivan, 2017).

Salah satu inovasi teknologi yang bisa diterapkan dalam *smart home* adalah *human activity recognition* atau pengenalan aktivitas manusia. Teknologi pengenalan aktivitas manusia memungkinkan sebuah sistem mendeteksi aktivitas sederhana yang dilakukan manusia, seperti berdiri, duduk, berbaring, berjalan, berlari dan lain-lain menggunakan kamera atau sensor. Pengenalan aktivitas manusia banyak diaplikasikan untuk perawatan kesehatan seperti memantau aktivitas pasien yang sedang menjalani terapi (Jain & Kanhangad, 2018). Dalam penerapannya, pengenalan aktivitas manusia pada *smart home* bisa digunakan untuk mengontrol peralatan elektronik rumah berdasarkan aktivitas sederhana yang bisa dikenali oleh sistem.

Sistem untuk mendeteksi aktivitas manusia pernah diteliti menggunakan kamera yang akan mengambil gambar berupa citra yang kemudian diproses untuk mengenali aktivitas yang sedang dilakukan. Metode pengenalan aktivitas manusia berbasis citra mempunyai kekurangan yaitu tidak adaptif terhadap cahaya, sehingga akurasi sistem menurun apabila pencahayaan ruangan terlalu gelap atau terlalu terang (Putra, 2016).

Penelitian pengenalan aktivitas manusia yang lain menggunakan sensor *accelerometer* dan *gyroscope* yang ditempatkan pada beberapa bagian tubuh. Sistem ini tergolong teknologi *wearable device*, dimana sistem yang dibuat dapat dikenakan pada tubuh pengguna. Penelitian yang terkait dengan penelitian ini telah dilakukan oleh Muhammad Zubair yang menggunakan 4 buah sensor *accelerometer* yang ditempatkan pada 4 bagian tubuh yang berbeda, yaitu pinggang, paha kiri, pergelangan kaki kanan dan lengan kanan serta menggunakan metode klasifikasi *Random forest* dan *decision tree* (Zubair, 2016). Namun, penggunaan sensor yang cukup banyak pada perangkat *wearable*, dapat menyebabkan ketidaknyamanan dalam pemakaian perangkat (Cao, et al., 2017). Selain itu, penggunaan sensor yang cukup banyak juga dapat mengakibatkan masa penggunaan baterai yang semakin cepat habis (Cao, et al., 2017).

Berdasarkan latar belakang yang telah diuraikan, penulis mengajukan penelitian mengenai sistem yang dapat mendeteksi aktivitas manusia menggunakan sensor *accelerometer* dan sensor *gyroscope* yang dipasang pada

manusia dan akan diklasifikasi menggunakan metode *k-Nearest Neighbor*. Sistem ini diharapkan bisa mendeteksi aktivitas manusia dengan tidak tergantung pada pencahayaan ruangan. Sensor yang digunakan adalah sensor MPU6050, dimana sensor ini sudah terintegrasi dengan sensor *accelerometer* dan *gyroscope*. sistem ini menggunakan Arduino sebagai pengendali utama sistem dan modul komunikasi wireless NRF24L01 untuk pengiriman dan penerimaan data sensor. Selain itu, pada penelitian ini akan diimplementasikan sistem yang menggunakan satu sensor dan sistem yang menggunakan dua sensor, sehingga diharapkan dapat menghasilkan sistem yang mendapatkan akurasi baik dengan sensor yang sedikit mungkin.

Prinsip kerja sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* adalah dengan mengklasifikasi hasil keluaran sensor tersebut ke dalam beberapa aktivitas yang ditentukan, yaitu berdiri, duduk, berbaring dan berjalan. Klasifikasi dilakukan menggunakan metode *k-Nearest Neighbor*. Metode klasifikasi *k-Nearest Neighbor* digunakan karena salah satu kelebihan metode ini yaitu tangguh terhadap *training* data yang *noisy* sesuai dengan karakteristik hasil keluaran sensor MPU6050 yang memiliki banyak *noise* (Sukma, et al., 2014).

1.2 Rumusan masalah

1. Bagaimana merancang sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode *k-Nearest Neighbor*?
2. Bagaimana mengimplementasikan sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode *k-Nearest Neighbor*?
3. Bagaimana hasil akurasi sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode *k-Nearest Neighbor*?
4. Berapa lama waktu komputasi metode *k-Nearest Neighbor* dalam melakukan klasifikasi aktivitas manusia?

1.3 Tujuan

1. Untuk merancang sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode *k-Nearest Neighbor*.
2. Untuk mengimplementasikan sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode *k-Nearest Neighbor*.
3. Untuk mengetahui hasil akurasi sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode *k-Nearest Neighbor*.
4. Untuk mengetahui waktu yang dibutuhkan metode *k-Nearest Neighbor* dalam melakukan klasifikasi aktivitas manusia.

1.4 Manfaat

Dalam penyelesaiannya, diharapkan penelitian ini dapat memberikan kontribusi serta manfaat untuk berbagai pihak, baik untuk peneliti maupun untuk pengguna atau pembaca. Berikut manfaat yang diharapkan dari penelitian ini.

Bagi Peneliti:

Penelitian ini diharapkan dapat digunakan sebagai penerapan dari teori dan praktik dari mata kuliah yang telah dipelajari selama mengikuti pendidikan di Program Studi Teknik Komputer Fakultas Ilmu Komputer Universitas Brawijaya dengan mengimplementasikan sensor *gyroscope* dan *accelerometer* untuk sistem pengenalan aktivitas manusia menggunakan metode *K-Nearest Neighbor* untuk *Smart home*.

Bagi Pengguna/Pembaca:

Penelitian ini diharapkan dapat memberi manfaat bagi pengguna atau pembaca yang sedang mencari referensi dalam mengimplementasikan sensor *gyroscope* dan *accelerometer* untuk sistem pengenalan aktivitas manusia menggunakan metode *K-Nearest Neighbor* untuk *Smart home*.

1.5 Batasan masalah

1. Sistem hanya dapat mengklasifikasi 4 aktivitas sederhana, yaitu berdiri, duduk, berbaring dan berjalan.
2. Hasil klasifikasi ditampilkan pada *Serial Monitor* Arduino IDE.
3. Proses klasifikasi hanya bisa dilakukan jika semua nilai sensor sudah diterima oleh Arduino Mega.
4. Rangkaian sensor diletakkan pada bagian tubuh yang telah ditentukan.
5. Desain dan berat sistem tidak menjadi bahasan pada penelitian ini.

1.6 Sistematika pembahasan

Sistematika pembahasan dalam penelitian ini akan disusun sebagai berikut:

BAB I PENDAHULUAN

Bab I membahas mengenai latar belakang masalah, identifikasi masalah yang berdasar pada latar belakang, rumusan masalah yang digunakan sebagai acuan dalam melakukan penelitian, tujuan dilakukan penelitian, manfaat penelitian, batasan penelitian dan sistematika pembahasan.

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Bab II membahas penelitian yang sudah pernah dilakukan yang berhubungan dengan penelitian yang sedang diajukan serta sebagai rujukan dan referensi dalam melakukan penelitian dan dasar teori yang didapat dari sumber-sumber yang relevan untuk digunakan sebagai referensi dalam pembuatan proposal skripsi.

BAB III METODE PENELITIAN

Bab III membahas mengenai metode penelitian yang digunakan untuk merancang sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode *k-Nearest Neighbor* berbasis Arduino.

BAB IV REKAYASA KEBUTUHAN

Bab IV membahas mengenai analisis kebutuhan untuk perancangan dan implementasi sistem, yang terdiri dari analisis kebutuhan fungsional, analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak.

BAB V PERANCANGAN DAN IMPLEMENTASI

Bab V akan dipaparkan mengenai perancangan dan implementasi sistem, baik dari sisi perangkat keras maupun perangkat lunak.

BAB VI PENGUJIAN DAN ANALISIS

Bab VI akan dipaparkan mengenai proses pengujian dan hasil pengujian. Serta dilakukan analisis terhadap hasil pengujian.

BAB VII PENUTUP

Bab VI membahas mengenai kesimpulan dan saran yang didapatkan dari hasil penelitian dan pembahasan serta dibuat dengan mengacu pada rumusan masalah.

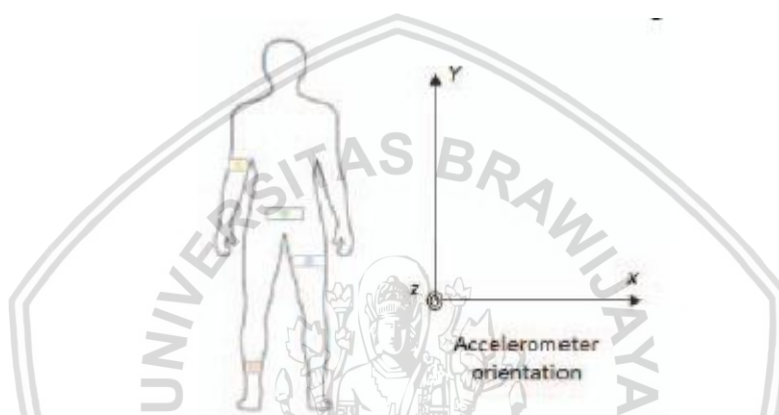
LAMPIRAN

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

2.1.1 Human Activity Recognition Using Wearable Accelerometer Sensor

Penelitian yang telah dilakukan oleh Muhammad Zubair mengenai pengenalan aktivitas manusia menggunakan sensor *wearable*. Penelitian ini menggunakan sensor 4 buah accelerometer yang diletakan pada 4 bagian tubuh yang berbeda, yaitu pinggang, paha kiri, pergelangan kaki kanan, dan lengan kanan. Ilustrasi gambar peletakan sensor terdapat pada Gambar 2.1 mengenai skema posisi sensor.



Gambar 2.1 Skema Posisi Sensor

Sumber: (Zubair, 2016)

Aktivitas yang diklasifikasi dalam penelitian ini adalah duduk, berdiri dan berjalan serta menggunakan metode klasifikasi *random forest* dan *decision tree*. Akurasi yang dihasilkan pada penelitian ini sebesar 99,9%. (Zubair, 2016)

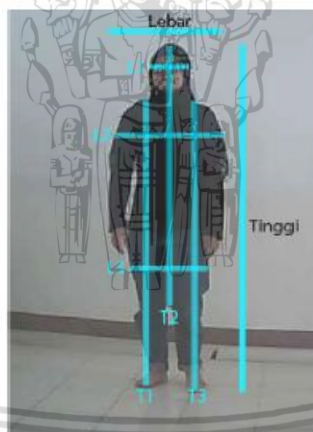
2.1.2 Rancang Bangun Sistem Monitoring Deteksi Jatuh untuk Manula dengan Menggunakan Accelerometer

Penelitian lain yang telah dilakukan oleh Siti Norhabibah yang mengimplementasikan sensor accelerometer untuk mendeteksi manula ketika terjatuh. Penelitian ini juga menggunakan modul sensor MPU6050 yang dirancang dengan arduino dan modul bluetooth HC-05. Pada penelitian ini, besaran akselerasi pada sumbu x, y, dan z dari sensor accelerometer diolah menjadi nilai total akselerasi. Kemudian nilai tersebut akan dibandingkan dengan nilai threshold yang lebih rendah dan nilai threshold yang lebih tinggi. Setelah itu, nilai total akselerasi akan dibandingkan lagi dengan nilai threshold dari nilai total orientasi yang digunakan untuk membedakan aktivitas biasa dan jatuh. Ketika aktifitas jatuh telah dideteksi, sistem kemudian mengirimkan notifikasi kepada user melalui aplikasi mobile.

Dari hasil pengujian yang telah dilakukan dalam mendeteksi jatuh sebanyak 150 kali, diperoleh nilai sensitifity sebesar 89% dan nilai specificity sebesar 98% yang berarti sistem pada penelitian ini mampu mendeteksi aktifitas jatuh dan aktifitas biasa dengan cukup baik. Sistem pada penelitian ini membutuhkan rata-rata 2 detik pasca aktifitas jatuh untuk mengirimkan notifikasi kepada user. Kekurangan dari sistem pada penelitian ini yaitu karena sistem menggunakan algoritma fall detection dengan memanfaatkan data pasca-impact sehingga notifikasi baru muncul rata-rata 2 detik setelah jatuh. (Norhabibah, 2016)

2.1.3 Identifikasi Aktivitas Manusia di Dalam Ruangan Menggunakan IP Camera dengan Metode Template Matching dan Algoritma Klasifikasi

Pada penelitian lain yang mengidentifikasi aktivitas manusia digunakan IP Camera untuk mendapatkan citra sebagai input yang kemudian akan diolah menggunakan metode template matching dan diklasifikasi menggunakan algoritma klasifikasi k-Nearest Neighbors. Variabel-variabel yang diklasifikasi adalah variabel seperti rasion tinggi dan lebar dari citra input. Aktivitas yang diidentifikasi pada penelitian ini adalah berdiri, duduk, jongkok dan berbaring. Penelitian ini juga direncanakan untuk diterapkan pada ruangan untuk memonitoring aktivitas lansia pada suatu ruangan. (Putra, 2016)



Gambar 2.2 Pola Objek Saat Berdiri

Setelah dilakukan pengujian, didapatkan tingkat akurasi sistem sebesar 80% untuk identifikasi 4 aktivitas, yaitu berdiri, duduk, jongkok dan berbaring. Pada penelitian ini didapatkan kesimpulan bahwa cahaya dapat mempengaruhi akurasi sistem dalam proses identifikasi aktivitas manusia, sementara jarak antara objek dengan IP Camera tidak terlalu mempengaruhi sistem dalam melakukan identifikasi. (Putra, 2016)

2.2 Dasar Teori

2.2.1 Aktivitas Manusia

Aktivitas manusia merupakan suatu kegiatan yang dilakukan oleh manusia. Aktivitas manusia dikelompokkan menjadi beberapa jenis, diantaranya aktivitas sederhana, aktivitas kompleks, aktivitas kesehatan, aktivitas hidup dan aktivitas kerja. Aktivitas sederhana yang dilakukan manusia seperti berjalan, duduk, berdiri, *jogging*, dan lain-lain. Kemudian terdapat aktivitas kompleks yang biasanya berhubungan dengan beberapa aktivitas dengan periode yang lebih lama, seperti naik bus, mengemudi, berbelanja, dan lain-lain. (Su, 2014)

Terdapat aktivitas yang berkaitan dengan kesehatan, seperti olahraga, jatuh, rehabilitasi, dan lain-lain. Selain itu terdapat beberapa aktivitas yang berhubungan dengan kehidupan, antara lain menyikat gigi, menyedot debu, memasak, mencuci piring, dan sebagainya. Kelompok aktivitas yang lain adalah aktivitas yang berhubungan dengan kerja, seperti membersihkan, rapat, dan lain-lain. Beberapa aktivitas hanya melibatkan bagian-bagian tubuh tertentu, seperti mengetik dan melambatkan tangan. (Su, 2014)

Selain itu, terdapat level aktivitas manusia yang dikategorikan berdasarkan kompleksitasnya. Pengelompokan ini berdasarkan jumlah orang yang terlibat dalam suatu kegiatan. Aktivitas manusia yang berada pada level *actions* merupakan aktivitas yang bisa dilakukan hanya dengan satu orang. Aktivitas tetap bisa dilakukan meskipun tanpa orang lain yang terlibat. Seperti berjalan, duduk, berdiri dan sebagainya. Aktivitas yang melibatkan dua orang untuk melakukannya merupakan aktivitas manusia level *interactions*. Misalnya meninju yang membutuhkan orang lain sebagai objek tinju, bersalaman dan sebagainya. Selain itu, terdapat aktivitas yang hanya bisa dilakukan dengan dua orang lebih atau group activities, misalkan menari kelompok. (Srikanth, et al., 2012)

2.2.2 Metode k-Nearest Neighbor

Metode k-Nearest Neighbor merupakan salah satu metode klasifikasi yang termasuk dalam algoritma *supervised learning* (Oktaviani, 2016). Supervised learning merupakan algoritma yang membutuhkan data latih atau data training untuk melakukan prediksi dan menentukan hasil klasifikasi (SM Team, 2017). Metode k-Nearest Neighbor juga disebut sebagai algoritma *instance-based learning*, dimana metode ini melakukan pembelajaran dalam menentukan hasil klasifikasi berdasarkan data yang sudah ada (Ndaumanu, 2014).

Metode k-Nearest Neighbor melakukan klasifikasi berdasarkan data latih yang jaraknya paling dekat dengan data yang baru. Metode ini melakukan prediksi dengan menghitung jarak data yang baru dengan seluruh data latih yang telah didefinisikan. Menghitung jarak antara data yang baru dengan data latih bisa menggunakan rumus *Euclidean Distance* (Sukma, et al., 2014). Rumus ini menghitung jarak antara dua data berdasarkan selisih antara data satu dengan data yang kedua. Rumus Euclidean Distance terdapat pada Persamaan 2.1.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 \dots \dots (n_1 - n_2)^2} \quad (2.1)$$

Pada persamaan 2.1 diketahui bahwa d merupakan jarak antara data latih dan data baru. Fitur-fitur data latih didefinisikan dengan variabel₁ yaitu x_1, y_1 hingga n_1 . Sedangkan fitur-fitur pada data yang baru didefinisikan dengan variabel₂, yaitu x_2, y_2 hingga n_2 .

Matode ini membutuhkan konstanta yang dilambangkan dengan k yang digunakan untuk menentukan jumlah tetangga terdekat yang akan dihitung kelas yang paling banyak muncul. Nilai k didapatkan dari hasil *trial-error*, yaitu dengan melakukan uji coba nilai k terhadap data uji hingga didapatkan nilai k dengan akurasi tertinggi. Data latih diurutkan berdasarkan perhitungan *Euclidean Distance* yang paling kecil hingga terbesar. Data yang sudah diurutkan diambil k data pertama yang akan dilakukan perhitungan kelas yang paling banyak muncul. Kelas yang paling banyak muncul akan dijadikan hasil klasifikasi.

Berikut merupakan langkah-langkah yang perlu dilakukan untuk melakukan klasifikasi menggunakan metode *k-Nearest Neighbor*. Langkah-langkah ini harus dilakukan berurutan dari poin pertama hingga terakhir.

1. Sebelum memulai klasifikasi, terlebih dahulu mendefinisikan data latih yang digunakan, karena metode ini tergolong algoritma *supervised learning* yang membutuhkan data latih untuk proses klasifikasi.
2. Kemudian menentukan parameter nilai k yang digunakan untuk menentukan jumlah tetangga terdekat.
3. Selanjutnya menghitung jarak *Euclidean* dari data uji berdasarkan data latih yang sudah ada.
4. Mengurutkan data berdasarkan hasil perhitungan jarak tersebut dari jarak *euclidean* terkecil hingga terbesar.
5. Mengambil dan membuat pengelompokan untuk data pertama hingga data sebanyak k yang ditentukan.
6. Data kelas yang paling banyak muncul pada kelompok data k akan menjadi hasil klasifikasi.

2.2.3 Sensor Accelerometer

Accelerometer merupakan perangkat elektromekanik yang akan mengukur gaya akselerasi atau nilai percepatan suatu benda. Gaya pada accelerometer ada yang bersifat statis dan dinamis. Gaya statis pada accelerometer disebabkan oleh gaya gravitasi yang menarik suatu benda. Sedangkan yang dinamis disebabkan oleh pergerakan atau pergetaran sensor accelerometer. Pada umumnya sensor accelerometer memanfaatkan efek piezoelectric atau dengan merasakan perubahan kapasitansi. (Berlia & Santosh, 2014)

Prinsip kerja accelerometer sesuai dengan namanya, yaitu prinsip percepatan (*acceleration*). Di dalam *accelerometer* terdapat per dengan beban yang dilepaskan. Beban tersebut bergerak dengan suatu kondisi tertentu kemudian

berhenti. Jika terdapat sesuatu yang membuat accelerometer bergerak atau tergoncang, maka beban tersebut akan berayun kembali. Pengukuran kapasitansi ini yang umumnya menjadi hasil pengukuran chip accelerometer. Agar sensor accelerometer bisa mendeteksi pergerakan tiga dimensi, maka dibutuhkan tiga pasang plat yang dipasang tegak lurus antar masing-masing. (Immersa, 2018)

2.2.4 Sensor Gyroscope

Sensor gyroscope adalah perangkat elektromekanik yang digunakan untuk mengukur atau mempertahankan orientasi. Gyroscope menggunakan prinsip ketetapan momentum sudut. Mekanisme kerja gyroscope adalah sebuah roda yang berputar dengan piringan di dalamnya yang tetap stabil. Sensor gyroscope menentukan orientasi gerak dengan bertumpu pada cakram atau roda yang berotasi dengan cepat pada suatu sumbu. Fungsi sensor gyroscope adalah untuk mendeteksi suatu gerakan sesuai dengan gravitasi, sehingga bisa dikatakan mendeteksi gerakan suatu benda. (Immersa, 2018)

Tegangan output dihasilkan ketika gyroscope bergerak. Pada kondisi diam, tegangan output yang dihasilkan akan bernilai konstan (Elektroku, 2017). Gyroscope memiliki keluaran nilai berupa kecepatan sudut dari tiga arah sumbu, yaitu sumbu x, y dan z. Nilai pada sumbu x akan menjadi sudut phi (pergerakan ke kanan dan kiri), nilai sumbu y menjadi sudut theta (pergerakan ke atas dan bawah) dan sumbu x menjadi sudut psi (pergerakan ke depan dan belakang) (Immersa, 2018).

2.2.5 MPU-6050

MPU-6050 merupakan modul sensor yang di dalamnya sudah terdapat *accelerometer MEMS (micro-electromechanical system)* dan *gyroscope MEMS (micro-electromechanical system)* dalam satu chip. MPU-6050 mengandung 16-bit analog yang dikonversi ke digital dalam setiap sumbu, sehingga menjadikan sensor ini semakin akurat. Sensor ini dapat mengambil nilai sumbu x, y, dan z secara bersamaan. Untuk koneksi dengan Arduino, sensor ini menggunakan antarmuka atau *interface I2C-bus (Inter Integrated Circuit)*. (Invensense, 2012)



Gambar 2.3 Modul Sensor MPU-6050

Sumber: (Arduino, 2016)

Gambar 2.3 menunjukkan sensor MPU-6050 jenis GY-521. MPU-6050 bisa digunakan untuk menghasilkan *output* “*Digital Motion Processor*” (DMP), juga disebut “*Processing Unit Motion Digital*”. DMP dapat melakukan perhitungan yang kompleks dengan nilai-nilai sensor yang dihasilkan MPU-6050. Perhitungan DMP dapat menggunakan *firmware* yang sudah disediakan. DMP dapat melakukan perhitungan langsung pada chip, sehingga membuat kerja mikrokontroler lebih ringan. (Arduino, 2016)

Untuk membaca nilai mentah atau *raw values* untuk *accelerometer* dan *gyroscope* sangat mudah. *Sleep mode* pada MPU-6050 harus dinonaktifkan, dan kemudian register untuk nilai *accelerometer* dan *gyroscope* dapat dibaca. Modul sensor ini juga berisi *buffer* FIFO 1024 byte. Nilai-nilai sensor dapat diprogram untuk ditempatkan pada *buffer* FIFO dan *buffer* dapat dibaca oleh Arduino. (Arduino, 2016)

2.2.6 Arduino Nano

Arduino Nano merupakan salah satu varian produk board mikrokontroler dari Arduino. Arduino Nano menggunakan mikrokontroler Atmega328 untuk versi 3.x dan Atmega168 untuk versi 2.x. Arduino jenis ini merupakan board mikrokontroler Arduino terkecil dengan ukuran PCB 18 x 45 mm dan berat 7 gram. Gambar Arduino Nano ditunjukkan pada Gambar 2.4.



Gambar 2.4 Layout board Arduino Nano

Sumber: (Arduino, 2016)

Arduino Nano memiliki fungsi yang kurang lebih sama dengan Arduino Duemilanove, tetapi dalam *package* yang berbeda. Board ini dapat diberikan daya melalui koneksi USB Mini-B, catu daya eksternal *unregulated* dengan tegangan 6 - 20 volt (pin 30) atau catu daya eksternal *regulated* dengan tegangan 5 volt (pin 27). Sumber daya dipilih ke sumber tegangan dipilih secara otomatis. Pada board Arduino Nano yang menggunakan Atmega 328, terdapat memori sebesar 32 KB, dimana 2 KB digunakan untuk *bootloader*, SRAM sebesar 2 KB dan EEPROM sebesar 1 KB. Memori ini berasal dari IC Atmega328. (Arduino, 2016)

Spesifikasi Arduino Nano versi 3.x terdapat pada Tabel 2.1.

Tabel 2.1 Spesifikasi Arduino Nano 3.x

Mikrokontroler	Atmega328
Arsitektur	AVR
Tegangan Operasi	5V
Memori Flash	32 KB, 2 KB digunakan untuk <i>bootloader</i>
SRAM	2 KB
Kecepatan clock	16 MHz
Pin I/O Analog	8
EEPROM	1 KB
Arus DC untuk Pin I/O	40 mA
Tegangan <i>Input</i>	7 - 12 Volt
Pin I/O digital	22
Output PWM	6
Konsumsi Daya	19 mA
Ukuran PCB	18 x 45 mm
Berat	7 g
Kode Produk	A000005

Sumber: (Arduino, 2016)

Dari 22 pin digital pada Arduino Nano, 14 pin diantaranya dapat digunakan sebagai input atau output menggunakan fungsi `pinMode()`, `digitalWrite()` dan `digitalRead()`. Pin-pin tersebut beroperasi pada tegangan 5 volt dan setiap pin dapat memberikan atau menerima arus maksimum sebesar 40 mA serta memiliki resisto *pull-up* internal (terputus secara *default*) 20-50 kOhm. Selain itu, beberapa pin memiliki fungsi khusus sebagai berikut (Arduino, 2016):

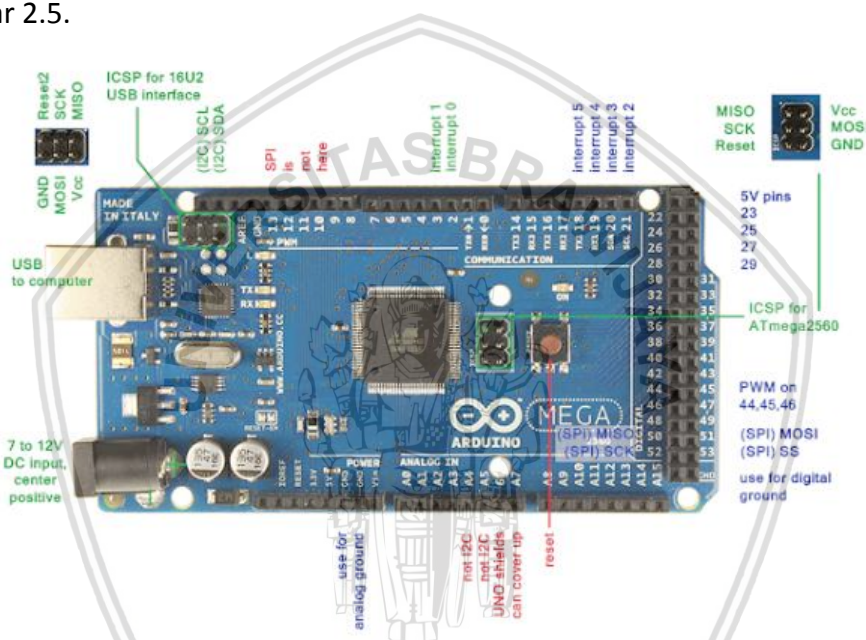
- Serial: Pin 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pin ini terhubung dengan pin yang sesuai dari chip FTDI USB-to-Serial TTL.
- *External Interrupt*: Pin 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai rendah, tapi naik atau turun, atau perubahan nilai.
- PWM (*Pulse Width Modulation*): Pin 3, 5, 6, 9, 10 dan 11. Pin ini menyediakan output PWM 8-bit dengan fungsi `analogWrite()`.
- SPI (*Serial Peripheral Interface*): Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI.
- LED: Pin 13. Pada Arduino Mega terdapat built-in LED yang terhubung ke pin digital 13. Ketika pin bernilai *HIGH*, LED akan menyala, ketika bernilai *LOW*, LED akan mati.

Selain itu, Arduino Nano juga memiliki 8 input analog, yang masing-masing menyediakan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara default,

pengukuran dilakukan dari *ground* hingga 5 volt. Pin analog 6 dan 7 tidak dapat digunakan sebagai pin digital. Beberapa pin analog Arduino Nano memiliki fungsi khusus, yaitu I2C yang terdapat pada pin A4 (SDA) dan A5 (SCL) yang mendukung komunikasi I2C (TWI) menggunakan *library Wire* (Arduino, 2016) .

2.2.7 Arduino MEGA 2560

Board Arduino Mega 2560 adalah sebuah board arduino yang menggunakan IC mikrokontroler Atmega2560. Board ini memiliki Pin I/O yang relatif banyak, yaitu 54 digital Input/Output, 15 buah diantaranya dapat digunakan sebagai output PWM, 16 buah analog input, 4 UART. Arduino Mega 2560 dilengkapi kristal 16 MHz untuk menggunakan relatif sederhana tinggal menghubungkan power dari USB ke PC / Laptop (Arduino, 2016). Board Arduino Mega ditunjukkan pada Gambar 2.5.



Gambar 2.5 Arduino Uno

Sumber: (Arduino, 2016)

Arduino Mega dapat diaktifkan dengan koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Daya eksternal (non-USB) dapat berasal dari adaptor AC-ke-DC atau baterai. Board Arduino Mega dapat beroperasi pada daya eksternal 5.5 volt hingga 16 volts. Jika diberikan daya kurang dari 7 volt, pin 5 volt mungkin dapat memasok kurang dari 5 volt dan board Arduino mungkin tidak stabil. Jika menggunakan daya lebih dari 12 volt, pengatur tegangan dapat menjadi panas dan merusak board Arduino Mega. Sehingga kisaran daya yang disarankan adalah 7 volt hingga 12 volt. Board Arduino Mega memiliki memori flash sebesar 256 KB untuk menyimpan kode, dimana 8 KB digunakan untuk bootloader, 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan *library EEPROM*). Spesifikasi Arduino Mega 2560 ditunjukkan pada Tabel 2.2. (Arduino, 2016)

Tabel 2.2 Spesifikasi Arduino Mega 2650

Mikrokontroler	Atmega2560
Tegangan Operasi	5V
Tegangan Input (direkomendasikan)	7-12V
Tegangan Input (Batas)	6-20V
Pin I/O Digital	54 (15 diantaranya bisa digunakan untuk output PWM)
Pin I/O PMW Digital	15
Pin Analog <i>Input</i>	16
Arus DC tiap Pin I/O	20 mA
Arus DC untuk Pin 3.3V	50 mA
Memory <i>Flash</i>	256 KB
	8 KB digunakan oleh <i>bootloader</i>
SRAM	8 KB
EEPROM	4 KB
Kecepatan <i>Clock</i>	16MHz
Panjang	101.52 mm
Lebar	53.3 mm
Berat	36 g

Sumber: (Arduino, 2016)

Arduino Mega memiliki beberapa pin yang digunakan untuk memasok daya agar board Arduino Mega bisa diaktifkan. Pin daya pada Arduino Mega adalah sebagai berikut (Arduino, 2016):

- Pin VIN, merupakan tegangan input untuk board Arduino ketika menggunakan sumber daya eksternal (dibandingkan dengan 5 volt dari koneksi USB atau sumber daya teregulasi lainnya). Pin ini digunakan jika memasok tegangan melalui colokan listrik.
- Pin 5V. Pin ini menghasilkan tegangan sebesar 5 volt yang diatur dari regulator pada board.
- Pin 3V3. Pin ini memasok tegangan sebesar 3,3 volt yang dihasilkan oleh regulator board. Daya tarik arus maksimum sebesar 50 mA.
- GND, merupakan pin ground.
- IOREF. Pada board Arduino Mega pin ini menyediakan referensi tegangan yang dioperasikan oleh mikrokontroler.

Setiap pin dari 50 pin digital pada *board* Arduino Mega dapat digunakan sebagai input atau output menggunakan fungsi `pinMode()`, `digitalWrite()` dan `digitalRead()` yang beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan dan menerima arus maksimum 40 mA dan memiliki resistor *pull-up* internal. Selain itu, beberapa pin memiliki fungsi khusus sebagai berikut (Arduino, 2016):

- Serial: Pin 0 (RX), 1 (TX), 19 (RX), 18 (TX), 17 (RX), 16 (TX), 15 (RX), 14 (TX). Pin ini digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL.
- *External Interrupt*: Pin 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3) dan 21 (interrupt 2). Pin ini dapat dikonfigurasi untuk memicu interrupt pada nilai rendah, tepi naik atau turun, atau perubahan nilai.
- PWM (*Pulse Width Modulation*): Pin 2 hingga 13 dan 44 hingga 46. Pin ini menyediakan *output* PWM 8-bit dengan fungsi `analogWrite()`.
- SPI (*Serial Peripheral Interface*): Pin 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin ini mendukung komunikasi SPI menggunakan *library* SPI.
- USB Host: MAX3421E

Board Arduino Mega memiliki 16 input analog, masing-masing menyediakan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara *default*, perhitungan analog dimulai dari *ground* hingga 5 volt, meskipun mungkin untuk mengubah ujung atas jangkauan, digunakan pin AREF dan fungsi `analogReference()`. (Arduino, 2016)

2.2.8 NRF24L01

Modul Wireless nRF24L01 adalah sebuah modul komunikasi jarak jauh yang memanfaatkan pita gelombang RF 2.4GHz ISM (*Industrial, Scientific and Medical*). Modul ini menggunakan antarmuka SPI untuk berkomunikasi. Tegangan kerja dari modul ini adalah 5 volt DC. Modul ini memiliki 8 buah pin, yaitu VCC (3,3 volt), Ground, CE, CSN, MOSI, MISO, SCK dan IRQ. (Dejan, 2017)



Gambar 2.6 Modul Wireless NRF24L01

Sumber: (Dejan, 2017)

NRF24L01 memiliki baseband logic Enhanced ShockBurst™ hardware protocol accelerator yang support “high-speed SPI interface for the application controller”. nRF24L01 memiliki true ULP solution, yang memungkinkan daya tahan baterai berbulan-bulan hingga bertahun-tahun. Modul ini dapat digunakan untuk pembuatan peripheral PC, piranti permainan, piranti fitness dan olahraga, mainan anak-anak dan alat lainnya. (Dejan, 2017)

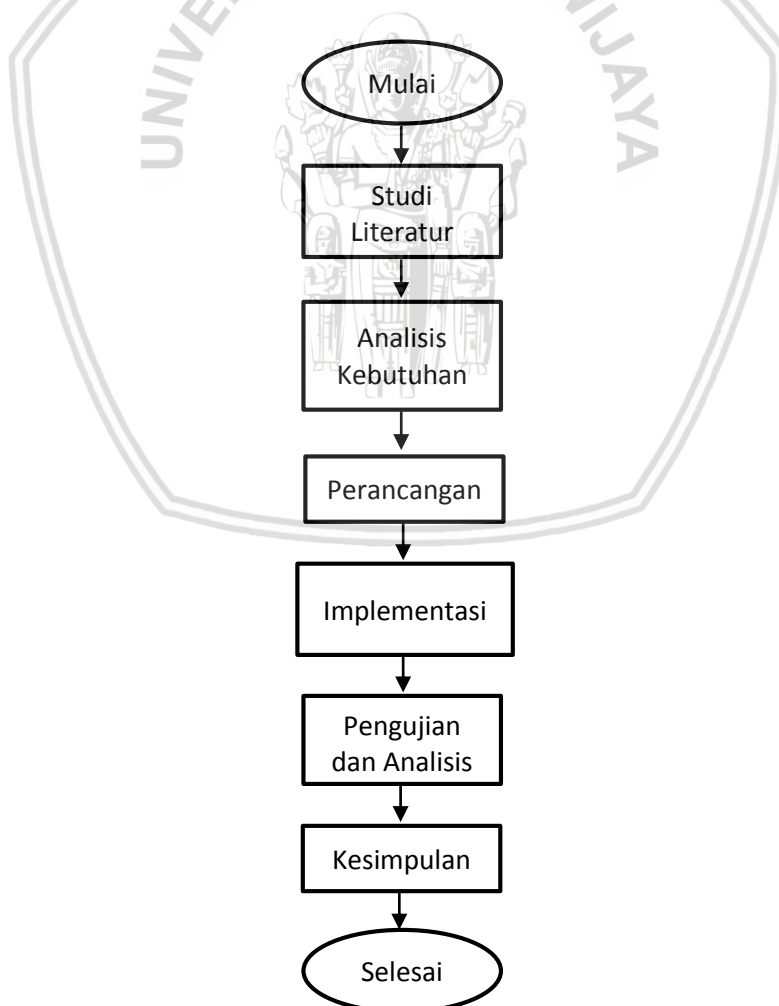
BAB 3 METODOLOGI

Bab ini menjelaskan metode penelitian yang digunakan. Metode penelitian merupakan langkah-langkah yang dilakukan secara sistematis untuk melakukan suatu penelitian. Setiap kegiatan penelitian akan dijelaskan pada masing-masing subbab.

3.1 Metode Penelitian

Tipe penelitian ini adalah Implementatif - Pengembangan (*Development*) karena penelitian ini membahas mengenai implementasi sistem klasifikasi aktivitas manusia menggunakan sensor accelerometer dan gyroscope dengan metode *k-Nearest Neighbor* Berbasis Arduino. Sehingga penelitian ini membutuhkan perancangan dan pengimplementasian perangkat keras dan perangkat lunak sehingga bisa digunakan untuk klasifikasi aktivitas manusia.

Gambar 3.1 menunjukkan alur metode penelitian yang digunakan pada penelitian ini sebagai acuan tindakan dalam melakukan penelitian.



Gambar 3.1 Diagram Alir Metode Penelitian

Langkah penelitian ini dimulai dari studi literatur dengan mempelajari berbagai literatur yang sesuai dengan latar belakang, rumusan masalah dan tujuan penelitian ini. Kemudian dilakukan analisis kebutuhan untuk menganalisis seluruh kebutuhan sistem agar bisa dilakukan perancangan dan pengimplementasian sistem. Setelah itu dilakukan pengujian dan analisis untuk mengetahui performansi sistem. Setelah semua proses tersebut, penarikan kesimpulan dan saran bisa dilakukan.

3.2 Studi Literatur

Sebelum dilakukan penelitian, studi literatur diperlukan untuk mendapatkan dan mempelajari penelitian-penelitian yang pernah dilakukan yang masih berkaitan dengan penelitian ini. Studi literatur mengenai teori-teori pendukung seperti metode yang digunakan atau komponen-komponen hardware yang digunakan juga diperlukan untuk mendapatkan gambaran umum tentang teori-teori tersebut. Studi literatur dari penelitian ini diambil dari beberapa sumber seperti jurnal atau paper yang diterbitkan dari hasil konferensi ilmiah, buku elektronik atau buku cetak, serta referensi internet dengan sumber yang jelas. Berikut beberapa teori dan literatur yang digunakan dalam penelitian ini.

1. Aktivitas Manusia
2. Metode *k-Nearest Neighbor*
3. Sensor *Accelerometer*
4. Sensor *Gyroscope*
5. Modul Sensor MPU-6050
6. Modul Wireless NRF24L01
7. Arduino Nano
8. Arduino Mega 2560

3.3 Analisis Kebutuhan

Analisis kebutuhan diperlukan untuk menganalisis seluruh kebutuhan untuk perancangan dan pengimplementasian sistem klasifikasi aktivitas manusia menggunakan sensor *gyroscope* dan *accelerometer* dengan metode *k-Nearest Neighbor* berbasis Arduino. Analisis kebutuhan pada penelitian ini meliputi kebutuhan fungsional sistem, kebutuhan perangkat keras dan kebutuhan perangkat lunak.

Kebutuhan fungsional sistem merupakan layanan-layanan atau fitur-fitur yang harus dimiliki sistem agar dapat bekerja sesuai dengan tujuan pembuatan sistem. Kebutuhan fungsional pada penelitian ini adalah sebagai berikut:

1. Sistem dapat membaca nilai sensor *accelerometer* dan *gyroscope*.
2. Sistem dapat mengirim dan menerima data sensor menggunakan modul NRF24L01.
3. Sistem dapat mengklasifikasi data sensor menggunakan metode *k-Nearest Neighbor* ke dalam empat kelas, yaitu berdiri, duduk, berbaring dan berjalan.

4. Sistem dapat menampilkan hasil klasifikasi pada Serial Monitor Arduino IDE.

Selain itu juga terdapat kebutuhan perangkat keras dan kebutuhan perangkat lunak. Kebutuhan perangkat keras dibutuhkan untuk menganalisis kebutuhan komponen - komponen perangkat keras untuk membangun sistem klasifikasi aktivitas manusia. Analisis kebutuhan perangkat keras didapatkan dari hasil studi literatur. Kebutuhan perangkat keras dalam penelitian ini adalah sebagai berikut:

1. Sensor MPU6050
2. Modul NRF24L01
3. Arduino Nano
4. Arduino Mega
5. Batrei

Sedangkan analisis kebutuhan perangkat lunak diperlukan untuk menganalisis apa saja perangkat lunak yang dibutuhkan untuk menjalankan komponen - komponen perangkat keras. Perangkat lunak yang digunakan harus sesuai dengan spesifikasi perangkat keras. Pada penelitian ini kebutuhan perangkat lunak yang digunakan adalah sebagai berikut:

1. Arduino IDE yang digunakan untuk menulis program sekaligus sebagai *compiler* program untuk Arduino.
2. Library Arduino yang diletakan pada bagian program untuk membantu komputasi program Arduino.

3.4 Perancangan

Perancangan dan implementasi sistem dibagi menjadi dua, yaitu perancangan dan implementasi perangkat keras dan perancangan dan implementasi perangkat lunak.

3.4.1 Perancangan Perangkat Keras

Perancangan dan implementasi perangkat keras bertujuan untuk membuat perangkat keras sistem klasifikasi aktivitas manusia. Penelitian ini menggunakan dua rangkaian perangkat keras, yaitu rangkaian perangkat keras untuk pengambilan data yang menggunakan sensor dan perangkat keras untuk menerima data dan untuk mengklasifikasi data yang telah diterima menggunakan metode *k-Nearest Neighbor*.

Rangkaian perangkat keras untuk akuisisi nilai sensor terdiri dari beberapa komponen, yaitu sensor MPU6050 yang digunakan untuk *sensing* atau mengambil data, Arduino Nano, modul *wireless* NRF24L01 dan sebuah batrei untuk memberikan daya pada Arduino. Sensor MPU6050 sudah terintegrasi dengan sensor *accelerometer* dan *gyroscope*, sehingga output dari sensor ini terdiri dari output nilai *accelerometer* dan *gyroscope*. Arduino Nano digunakan sebagai otak dari sistem ini. Arduino Nano sebagai modul mikrokontroler akan diberikan perintah-perintah untuk menjalankan sensor dan komponen lainnya. Sedangkan

modul radio NRF24L01 berfungsi untuk mengirim data nilai sensor yang telah disimpan menggunakan komunikasi radio secara *wireless*.

Rangkaian perangkat keras untuk penerima data dan klasifikasi terdiri dari modul mikrokontroler Arduino Mega 2560 dan modul *wireless* NRF24L01. Kedua komponen perangkat keras tersebut digunakan untuk menerima data hasil pembacaan sensor dari rangkaian perangkat keras akuisisi data dan mengklasifikasi menggunakan metode *k-Nearest Neighbor*. Hasil klasifikasi sistem akan ditampilkan pada serial monitor Arduino IDE.

3.4.2 Perancangan Perangkat Lunak

Setelah dilakukan perancangan perangkat keras, perancangan perangkat lunak dilakukan untuk membuat merancang logika yang sesuai dengan tujuan dan kebutuhan sistem. Perangkat lunak sistem terdiri dari program untuk akuisisi data sensor dan klasifikasi. Setelah sensor melakukan akuisisi nilai sensor, data tersebut akan dikirimkan menggunakan komunikasi radio kepada penerima. Setelah data yang dikirimkan sudah lengkap, sistem akan mengklasifikasi data-data tersebut menggunakan metode *k-Nearest Neighbor*.

3.4.3 Perancangan Pengambilan Data

Pengambilan data yang digunakan sebagai data *input* klasifikasi dilakukan dengan membaca nilai sensor *accelerometer* dan *gyroscope*. Kedua sensor ini masing-masing memiliki komponen nilai x, y dan z. Pembacaan nilai sensor dilakukan tiga kali. Nilai sensor *accelerometer* dan *gyroscope* yang masih dalam bentuk data mentah atau *raw data* akan diproses menggunakan rumus nilai total *accelerometer* dan *gyroscope*. Pada penelitian ini, satu sensor MPU6050 menghasilkan 6 fitur yang akan dijelaskan sebagai berikut:

1. Fitur 1, yaitu nilai total sensor *accelerometer* pada pembacaan data pertama.
2. Fitur 2, yaitu nilai total sensor *gyroscope* pada pembacaan data pertama.
3. Fitur 3, yaitu nilai total sensor *accelerometer* pada pembacaan data kedua.
4. Fitur 4, yaitu nilai total sensor *gyroscope* pada pembacaan data kedua.
5. Fitur 5, yaitu nilai total sensor *accelerometer* pada pembacaan data ketiga.
6. Fitur 6, yaitu nilai total sensor *gyroscope* pada pembacaan data ketiga.

Keenam nilai fitur tersebut akan dijadikan sebagai data array dengan panjang 6 elemen yang kemudian dikirimkan menggunakan modul NRF24L01. Pada sistem yang menggunakan satu sensor, terdapat 6 fitur seperti yang sudah dijelaskan. Sedangkan pada sistem yang menggunakan dua sensor MPU6050, terdapat 12 fitur yang terdiri dari 6 fitur yang berasal dari sensor pertama dan 6 fitur yang berasal dari sensor kedua.

3.5 Implementasi

Pada implementasi sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode *k-Nearest Neighbor* berbasis

Arduino, terdapat beberapa tahap, yaitu implementasi perangkat keras, implementasi perangkat lunak, implementasi penentuan posisi peletakan sensor dan implementasi pengambilan data latih.

3.5.1 Implementasi Perangkat Keras

Implementasi dimulai dengan merangkai komponen - komponen perangkat keras sesuai dengan perancangan yang dirangkai di atas PCB. Komponen - komponen perangkat keras dihubungkan menggunakan kabel atau *jumper* dan di-*solder*. Kemudian dilakukan pengecekan fungsi perangkat keras. Jika semua fungsi perangkat keras yang telah dirangkai sudah sesuai, selanjutnya dilakukan implementasi perangkat lunak.

3.5.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak pada penelitian ini berupa penulisan kode program untuk ditanamkan pada Arduino Nano dan Mega. Program yang ditanamkan pada Arduino Nano merupakan program untuk pengambilan data sensor *accelerometer* dan *gyroscope* dan pengiriman data menggunakan modul NRF24L01. Sedangkan program yang ditanamkan pada Arduino Mega merupakan program untuk menerima data hasil pembacaan sensor dan melakukan klasifikasi menggunakan metode *k-Nearest Neighbor*. Beberapa *library* Arduino juga digunakan untuk membantu proses pembuatan program.

3.5.3 Implementasi Penentuan Posisi Peletakan Sensor

Pada tahap implementasi sistem, terdapat implementasi penentuan lokasi peletakan sensor. Pengujian dilakukan menggunakan satu sensor dan dua sensor. Sensor yang digunakan pada penelitian ini bersifat *wearable*, sehingga bisa dikenakan atau diletakkan pada bagian tubuh seseorang. Letak sensor pada penelitian adalah terletak pada beberapa bagian tubuh manusia, yaitu di kepala, lengan, pinggang, paha dan kaki bagian bawah. Pengujian yang menggunakan satu sensor terdapat 5 letak sensor dan pengujian yang menggunakan 2 sensor terdapat 10 kombinasi peletakan sensor yang didapatkan dari perhitungan kombinasi angka 5 terhadap 2.

3.5.4 Implementasi Pengambilan Data Latih

Pada tahap implementasi juga dilakukan pengumpulan data latih. Data latih merupakan data yang digunakan sebagai acuan sistem dalam menentukan keputusan dan melakukan klasifikasi. Prosedur pengambilan data latih adalah dengan melakukan percobaan langsung terhadap sensor. Sensor diletakan pada bagian tubuh yang telah ditentukan dan orang yang menggunakan sensor melakukan aktivitas berdiri, duduk, berbaring dan berjalan. Kemudian data hasil pembacaan sensor disimpan dan dimasukkan dalam program untuk dijadikan data latih.

3.6 Pengujian dan Analisis

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dirancang dan diimplementasi. Setelah itu dilakukan analisis pada setiap hasil pengujian. Beberapa skenario pengujian dijelaskan sebagai berikut:

3.6.1 Pengujian Fungsional Sistem

Pengujian fungsional sistem dilakukan untuk menguji fungsi sistem dari sisi perangkat keras dan perangkat lunak. Pengujian fungsional dilakukan sesuai dengan kebutuhan fungsional sistem.

3.6.1.1 Pengujian Pembacaan Nilai Sensor

Pengujian ini dilakukan untuk mengetahui apakah sensor yang digunakan dapat melakukan pembacaan nilai sensor. Skenario pengujian ini adalah dengan menampilkan hasil pembacaan sensor MPU6050 yang berupa data mentah dan data yang sudah diolah pada *Serial Monitor* Arduino IDE.

3.6.1.2 Pengujian Pengiriman dan Penerimaan Data

Tujuan pengujian ini adalah untuk mengetahui keberhasilan modul komunikasi wireless NRF24L01 dalam pengiriman dan penerimaan data. Skenario pengujian ini adalah dengan mengirimkan nilai pembacaan sensor *accelerometer* dan *gyroscope*. Terdapat dua sensor yang digunakan pada pengujian ini. Keduanya mengirimkan data pembacaan sensor menggunakan modul NRF24L01. Pengujian ini untuk mengetahui apakah terdapat data yang hilang dari pengiriman dan penerimaan data menggunakan modul NRF24L01.

3.6.1.3 Pengujian Tampilan pada Serial Monitor Arduino IDE

Pengujian ini dilakukan untuk mengetahui apakah *output* yang ditampilkan pada *Serial Monitor* Arduino IDE sudah sesuai dengan perancangan dan implementasi perangkat lunak atau program. Skenario pengujian ini dilakukan dengan menampilkan karakter sesuai yang ada pada program.

3.6.2 Pengujian Nilai K

Metode k-Nearest Neighbor membutuhkan nilai k, yaitu jumlah tetangga terdekat yang akan dicari kelas yang paling banyak muncul. Pada penelitian ini, nilai k didapatkan dari hasil pengujian nilai k terhadap 300 data dari 15 posisi peletakan sensor. Pengujian dilakukan dengan uji coba nilai k dari 1 sampai 10. Pada setiap nilai k yang diuji, akan dicari nilai keakuratannya. Nilai k yang mendapatkan akurasi paling tinggi akan diterapkan sebagai nilai k tetap pada program untuk klasifikasi aktivitas manusia dengan metode k-Nearest Neighbor.

3.6.3 Pengujian Akurasi Sistem Klasifikasi Aktivitas Manusia

Pengujian ini dilakukan untuk mendapatkan akurasi dari hasil klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode Naive Bayes. Skenario pengujian ini adalah dengan melakukan pengujian

sebanyak 4 kali pada orang yang berbeda. Setiap orang diminta untuk mengenakan sensor sesuai dengan peletakan yang telah ditentukan. Pada pengujian yang menggunakan satu sensor, terdapat 5 posisi peletakan sensor, yaitu di kepala, pinggang, lengan, paha dan kaki bagian bawah. Sedangkan pada pengujian yang menggunakan dua sensor, terdapat 10 kombinasi posisi peletakan sensor, yang didapatkan dari hasil operasi 5 kombinasi 2. Kemudian melakukan aktivitas-aktivitas sederhana yang akan diklasifikasikan pada penelitian ini, yaitu berdiri, duduk, berbaring dan berjalan. Setiap aktivitas dilakukan 5 kali pengambilan data, sehingga pada setiap orang dilakukan 20 kali pengambilan data. Total data pengujian yang didapatkan dari 4 orang pada setiap posisi peletakan sensor adalah 80 data uji. Tujuan dari pengujian dengan 4 orang yang berbeda yaitu untuk mengetahui apakah sistem klasifikasi aktivitas manusia mendapatkan hasil yang sama atau berbeda jika digunakan oleh orang yang berbeda.

3.6.4 Pengujian Waktu Komputasi Klasifikasi k-*Nearest Neighbor*

Tujuan pengujian waktu komputasi klasifikasi k-*Nearest Neighbor* adalah untuk mengetahui waktu yang dibutuhkan sistem untuk melakukan proses klasifikasi aktivitas manusia menggunakan metode k-*Nearest Neighbor*. Skenario pengujian ini adalah dengan mengukur waktu komputasi dari ketika data telah diterima hingga sistem menampilkan hasil klasifikasi pada *Serial Monitor* Arduino IDE.

3.7 Kesimpulan

Penarikan kesimpulan dilakukan setelah semua tahap dalam metode penelitian telah selesai dilakukan. Kesimpulan didapatkan dari seluruh tahap penelitian mulai dari studi literatur hingga pengujian dan analisis. Kesimpulan yang ditarik sesuai dengan latar belakang pada penelitian ini. Selain itu, terdapat saran untuk penelitian selanjutnya mengenai topik ini agar menghasilkan sistem yang lebih baik.

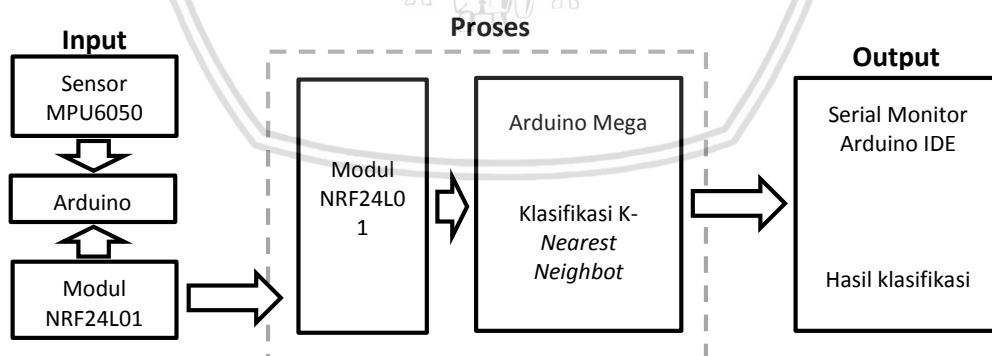
BAB 4 REKAYASA KEBUTUHAN

Pada bab ini dijelaskan secara terperinci mengenai rekayasa kebutuhan sistem sebelum dilakukan perancangan dan implementasi sistem. Pada bab ini juga dijelaskan mengenai gambaran umum sistem, lingkungan operasional dan batasan desain sistem.

4.1 Gambaran Umum Sistem

Sistem klasifikasi aktivitas manusia menggunakan sensor accelerometer dan gyroscope dengan metode k-Nearest Neighbor berbasis arduino ini merupakan sistem yang dapat mengklasifikasi aktivitas sederhana yang dilakukan manusia di rumah, yaitu berdiri, duduk, berbaring dan berjalan. Input dari sistem ini merupakan nilai sensor *accelerometer* dan *gyroscope* yang berupa data mentah dari komponen nilai x, y dan z yang telah diolah menjadi nilai total *accelerometer* dan *gyroscope*. Dalam penelitian ini digunakan sensor MPU6050 yang merupakan modul sensor yang di dalamnya sudah terdapat sensor *accelerometer* dan *gyroscope*. Sistem ini digunakan untuk mengenali dan mengklasifikasi aktivitas sederhana yang dilakukan di rumah, yaitu berdiri, duduk, berbaring dan berjalan. Pengembangan selanjutnya dari sistem ini dapat terintegrasi langsung dengan peralatan-peralatan elektronik pada *smarthome*.

Penelitian ini melakukan perancangan sistem klasifikasi aktivitas manusia menggunakan satu sensor dan 2 sensor. Pada sistem yang menggunakan satu sensor, hanya terdapat 6 fitur pada klasifikasinya. Sedangkan sistem yang menggunakan 2 sensor terdapat 12 fitur dimana masing-masing sensor memiliki 6 fitur. Diagram blok sistem klasifikasi aktivitas manusia yang menggunakan satu sensor terdapat pada Gambar 4.1.

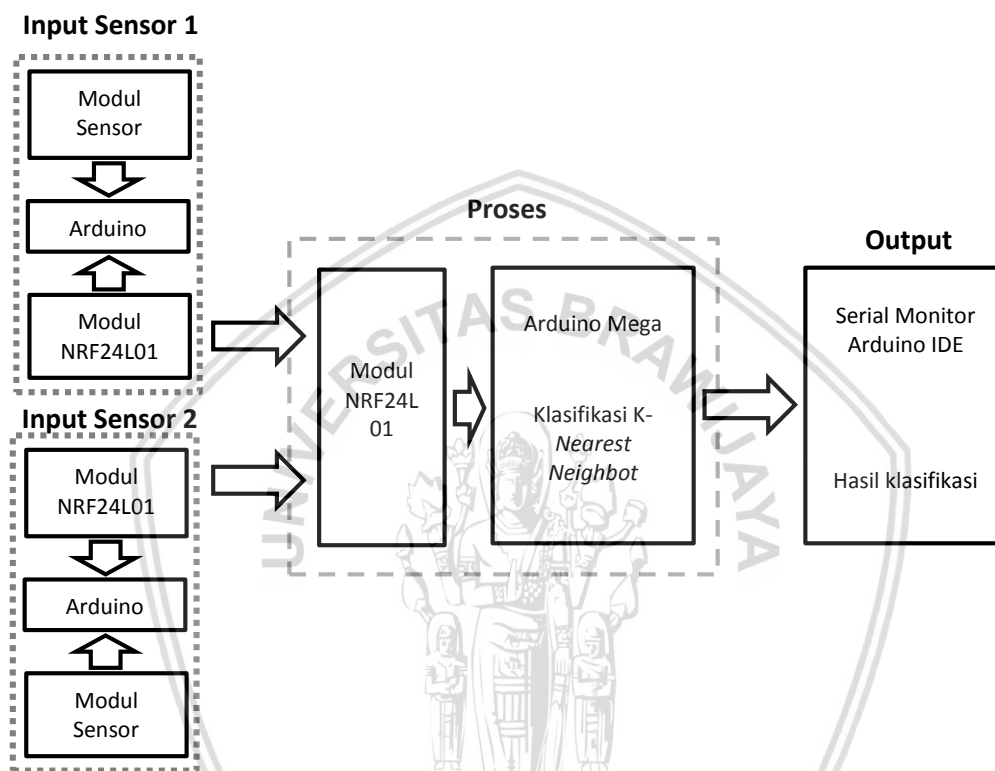


Gambar 4.1 Diagram Blok Sistem dengan Satu Sensor

Pada Gambar 4.1 dapat dilihat bahwa sistem yang menggunakan satu sensor, input hanya berasal dari satu sensor MPU6050, arduino nano bertindak sebagai mikrokontroler yang akan mengontrol input nilai sensor dan mengirimkan data nilai sensor menggunakan modul komunikasi *wireless* NRF24L01. Kemudian Arduino Mega yang terhubung dengan NRF24L01 akan menerima data nilai sensor

dan melakukan klasifikasi. Output sistem yang berupa hasil klasifikasi akan ditampilkan pada *Serial Monitor* Arduino IDE.

Sedangkan pada sistem yang menggunakan 2 sensor sebagai input untuk klasifikasi aktivitas manusia, terdapat 2 input yang berasal dari 2 sensor MPU6050. Masing-masing sensor dihubungkan dengan Arduino, sehingga terdapat 2 pengiriman yang akan diterima oleh Arduino Mega. Diagram blok sistem klasifikasi aktivitas manusia yang menggunakan satu sensor terdapat pada Gambar 4.2.



Gambar 4.2 Blok Diagram Sistem dengan 2 Sensor

Proses yang dilakukan pada sistem yang menggunakan 2 sensor sama dengan sistem yang menggunakan satu sensor, hanya berbeda pada data yang diterima oleh Arduino Mega. Proses klasifikasi hanya bisa dilakukan jika kedua nilai sensor sudah diterima oleh Arduino Mega.

4.1.1 Lingkungan Operasional Sistem

Pada penelitian ini, sistem hanya terbatas pada klasifikasi aktivitas manusia dimana hasil klasifikasi akan ditampilkan pada *Serial Monitor* Arduino IDE. Berikut merupakan persyaratan lingkungan operasional sistem untuk mendukung kinerja sistem:

1. Sensor MPU6050 secara bergantian diletakan pada beberapa bagian tubuh yang telah ditentukan, yaitu kepala, lengan, pinggang, paha dan kaki bagian bawah. Setiap peletakan pada bagian tubuh akan dihitung nilai akurasi masing-masing posisi peletakan sensor.

2. Pada sistem yang menggunakan 2 sensor, peletakan posisi sensor MPU6050 merupakan kombinasi dari 5 posisi pada bagian tubuh yang telah ditentukan.
3. Dalam setiap pengambilan data, dilakukan 3 kali pembacaan sensor dimana nilai sensor tersebut akan dijadikan data array.
4. Pengambilan data sensor hanya bisa dilakukan jika rangkaian sensor sudah terhubung dengan daya. Pada penelitian ini daya didapatkan dari baterai 9 volt yang dihubungkan dengan Arduino Nano.
5. Proses klasifikasi hanya bisa dilakukan jika semua data dari kedua sensor sudah diterima oleh Arduino Mega. Jika data dari salah satu sensor belum diterima, Arduino Mega akan terus melakukan penerimaan data.
6. Arduino Mega yang digunakan sebagai pengolah data untuk klasifikasi harus terhubung secara serial dengan komputer / laptop menggunakan kabel USB.

4.1.2 Batasan Desain Sistem

Batasan desain dalam perancangan dan pengimplementasian sistem klasifikasi aktivitas manusia adalah sebagai berikut:

6. Sistem hanya dapat mengklasifikasi 4 aktivitas sederhana, yaitu berdiri, duduk, berbaring dan berjalan.
7. Hasil klasifikasi ditampilkan pada *Serial Monitor* Arduino IDE.
8. Proses klasifikasi hanya bisa dilakukan jika semua nilai sensor sudah diterima oleh Arduino Mega.
9. Rangkaian sensor diletakkan pada bagian tubuh yang telah ditentukan.

4.1.3 Asumsi Ketergantungan

Beberapa asumsi ketergantungan sistem ini adalah sebagai berikut:

1. Pin sensor MPU6050 dan modul komunikasi *wireless* NRF24L01 terhubung dengan Arduino Nano dan Arduino Mega sesuai dengan perancangan sistem.
2. Jika sistem klasifikasi aktivitas manusia digunakan ketika melakukan aktivitas selain berdiri, duduk, berbaring dan berjalan, maka hasil output sistem adalah kelas yang memiliki data latih terdekat dengan nilai pembacaan sensor.
3. Proses klasifikasi hanya bisa dilakukan jika semua nilai sensor, baik dari sensor pertama atau sensor kedua sudah diterima oleh Arduino Mega.

4.2 Analisis Kebutuhan

Tahap analisis kebutuhan diperlukan untuk menganalisis seluruh kebutuhan untuk perancangan dan implementasi sistem deteksi aktivitas manusia. Analisis kebutuhan didapatkan dari hasil studi literatur. Dengan dilakukan analisis kebutuhan, diharapkan sistem dapat bekerja sesuai dengan tujuan pembuatan sistem. Kebutuhan yang dianalisis terdiri dari kebutuhan fungsional, kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan fungsi-fungsi atau layanan-layanan yang harus disediakan sistem agar sesuai dengan tujuan pembuatan sistem. Perancangan dan implementasi sistem dikatakan berhasil jika sebuah sistem bisa melakukan fungsi-fungsi tertentu yang diharapkan bisa dilakukan oleh sistem. Berikut merupakan kebutuhan fungsional sistem klasifikasi aktivitas manusia:

4.2.1.1 Sistem dapat Membaca Nilai Sensor *Accelerometer* dan *Gyroscope*

Sistem ini harus bisa membaca nilai sensor *accelerometer* dan *gyroscope* yang berupa nilai sumbu x, y dan z dan mengolah nilai tersebut menjadi total nilai *accelerometer* dan *gyroscope*. Fungsi ini sangat penting karena total nilai *accelerometer* dan *gyroscope* digunakan sebagai input sistem.

4.2.1.2 Sistem dapat Mengirim dan Menerima Data Sensor

Pada kebutuhan fungsional sistem ini, sistem harus bisa mengirim data nilai pembacaan sensor menggunakan modul komunikasi wireless NRF24L01 yang terhubung dengan Arduino Nano. Kemudian Arduino Mega sebagai pengklasifikasi harus bisa menerima data sensor yang telah dikirim.

4.2.1.3 Sistem dapat Mengklasifikasi Aktivitas Manusia dengan Metode k-*Nearest Neighbor*

Fungsi utama dari sistem ini adalah bisa melakukan klasifikasi aktivitas manusia berdasarkan data sensor *accelerometer* dan *gyroscope* dengan metode k-*Nearest Neighbor*. Sistem melakukan klasifikasi menjadi 4 kelas, yaitu berdiri, duduk, berbaring dan berjalan.

4.2.1.4 Sistem dapat Menampilkan Hasil Klasifikasi pada *Serial Monitor* Arduino IDE

Hasil klasifikasi aktivitas manusia ditampilkan pada *Serial Monitor* Arduino IDE, sehingga fungsi tersebut menjadi salah satu kebutuhan fungsional sistem yang harus dipenuhi. Hasil klasifikasi ditampilkan pada *Serial Monitor* agar bisa dilakukan analisis terhadap *output* sistem.

4.2.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak yang dijelaskan secara rinci pada subbab berikut.

4.2.2.1 Kebutuhan Perangkat Keras

Pada penelitian ini, diperlukan beberapa komponen perangkat keras untuk membangun sistem klasifikasi aktivitas manusia. Kebutuhan perangkat keras pada penelitian ini adalah sebagai berikut:

1. Sensor MPU6050

Sensor MPU6050 sudah terdapat sensor *accelerometer* dan *gyroscope* di dalamnya. Sensor ini digunakan sebagai *input* sistem.

2. Mikrokontroler Arduino Nano

Arduino Nano digunakan sebagai pemroses atau otak rangkaian yang terhubung dengan sensor MPU6050.

3. Mikrokontroler Arduino Mega

Arduino Mega digunakan sebagai penerima data sensor dan melakukan klasifikasi menggunakan metode *k-Nearest Neighbor*.

4. Modul NRF24L01

Modul NRF24L01 merupakan modul komunikasi *wireless* yang digunakan sebagai modul pengiriman data.

5. Baterai 9 volt

Baterai digunakan untuk memberikan daya listrik ke sistem. Baterai 9 volt dihubungkan langsung ke Arduino Nano.

6. Kabel *Jumper*

Kabel *jumper* digunakan untuk menghubungkan pin-pin antar komponen perangkat keras.

4.2.2.2 Kebutuhan Perangkat Lunak

Pada penelitian ini dibutuhkan perangkat lunak yaitu Arduino IDE yang digunakan untuk menuliskan program sekaligus mengoperasikan mikrokontroler Arduino. Dalam pembuatan program juga diperlukan beberapa *library* untuk membantu proses penulisan program, salah satunya *library* MPU6050.h dan RF24.h.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan mengenai perancangan dan implementasi sistem, diantaranya perancangan dan implementasi perangkat keras, perancangan dan implementasi perangkat lunak dan lain sebagainya. Masing-masing perancangan dan implementasi akan dijelaskan pada subbab berikut.

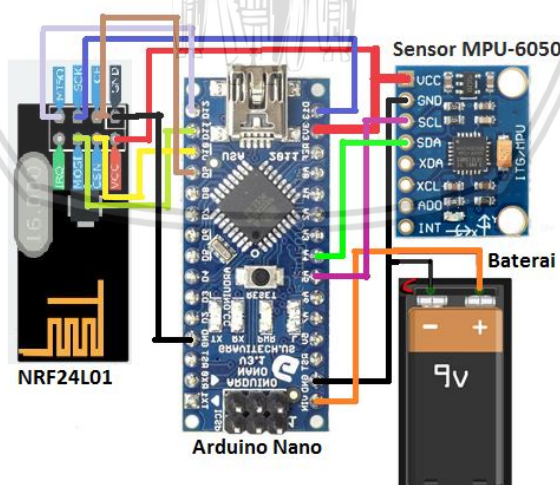
5.1 Perancangan Sistem

Pada subbab ini akan dijelaskan mengenai perancangan sistem, yang terdiri dari perancangan perangkat keras, perancangan perangkat lunak dan perancangan pengambilan data.

5.1.1 Perancangan Perangkat Keras

Pada sistem klasifikasi aktivitas manusia, terdapat 2 macam perangkat keras, yaitu perangkat keras untuk pengambilan data sensor dan pengiriman data, dan perangkat keras untuk menerima data dan melakukan klasifikasi.

Perancangan perangkat keras pada rangkaian pengambilan data ini terdiri dari rangkaian sensor, modul komunikasi, dan mikrokontroler. Mikrokontroler dirangkai terhubung dengan sensor dan modul komunikasi agar dapat membaca nilai sensor yang kemudian dikirim ke mikrokontroler yang lain menggunakan modul komunikasi tersebut. Terdapat 2 rangkaian perangkat keras pada sistem ini, yaitu rangkain node sensor untuk akuisisi nilai sensor dan rangkaian untuk mikrokontroler yang bertugas mengolah data dengan metode *K-Nearest Neighbor*. Adapun gambar skematik rangkaian ditunjukkan oleh Gambar 4.1. tentang gambar skematik rangkaian untuk akuisisi data.



Gambar 5.1 Rangkaian Skematik Akuisisi Data

Dari gambar rangkaian tersebut, terdapat 4 komponen perangkat keras. Arduino Nano digunakan sebagai pemroses yang berisi kode program untuk pengambilan dan pengiriman data. NRF24L01 merupakan modul komunikasi radio yang digunakan untuk mengirimkan data hasil pembacaan sensor ke perangkat

arduino yang lain. Modul NRF24L01 menggunakan komunikasi SPI (Serial Peripheral Interface) dengan arduino. Dari 8 pin yang terdapat pada NRF24L01, terdapat 7 pin yang digunakan untuk dihubungkan dengan Arduino Nano. Pin MISO pada NRF24L01 terhubung dengan pin digital 12 pada Arduino Nano, pin MOSI terhubung dengan pin digital 11 Arduino, pin SCK terhubung dengan pin digital 13, pin CE terhubung dengan pin digital 9, pin CSN terhubung dengan pin digital 10, pin GND dan VCC pada NRF24L01 terhubung dengan pin GND dan pin 3,3V pada Arduino Nano. Keterangan pin NRF24L01 dengan Arduino Nano lebih jelasnya terdapat pada tabel 5.1.

Tabel 5.1 Keterangan Pin NRF24L01 dengan Arduino Nano

Pin NRF24L01	Arduino Nano
MISO	D12
MOSI	D11
SCK	D13
CE	D9
CSN	D10
VCC	3V3
GND	GND

Pada modul sensor MPU6050, pin yang digunakan hanya 4 pin dari 8 pin yang tersedia di modul, yaitu pin VCC, GND, SCL dan SDA. Pin VCC pada MPU6050 terhubung dengan pin 3,3 volt pada Arduino Nano, pin GND MPU-6050 terhubung dengan pin GND Arduino Nano, pin SCL terhubung dengan pin A5, dan pin SDA terhubung dengan pin A4. Keterangan pin MPU6050 dengan Arduino Nano terdapat pada tabel 5.2.

Tabel 5.2 Keterangan Pin MPU6050 dengan Arduino Nano

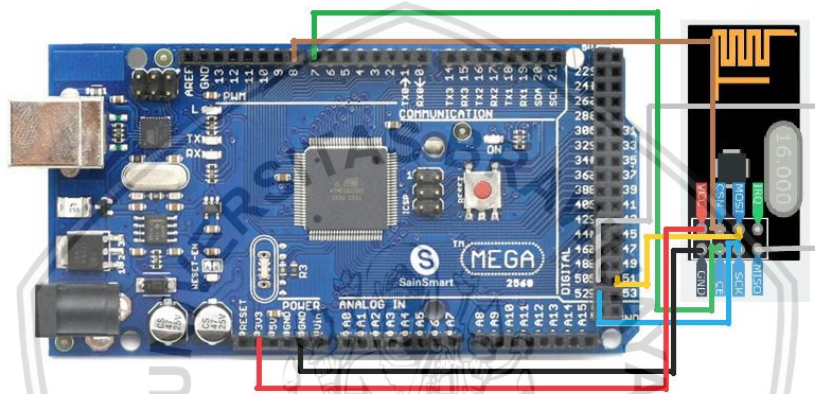
Pin MPU6050	Arduino Nano
VCC	3V3
GND	GND
SCL	A5
SDA	A4

Pada rangkaian tersebut juga terdapat baterai dengan daya 9 volt untuk memberi daya arduino nano. Pin Positif (+) baterai terhubung dengan pin Vin pada Arduino Nano dan pin Negatif (-) baterai terhubung dengan pin GND pada Arduino Nano. Keterangan pin baterai dengan Arduino Nano terdapat pada tabel 5.3.

Tabel 5.3 Keterangan Pin Arduino Nano dengan Baterai

Baterai	Arduino Nano
Positi (+)	Vin
Negatif (-)	GND

Pada sistem untuk klasifikasi aktivitas manusia ini, selain rangkaian perangkat keras untuk akuisisi data, juga terdapat rangkaian penerima dan pengolahan data yang terdiri dari Arduino Mega 2560 dan modul NRF24L01. Adapun gambar skematik rangkaiannya ditunjukkan oleh Gambar 5.2. tentang gambar skematik rangkaian penerima dan pengolah data.



Gambar 5.2 Gambar Skematik Rangkaian Penerima Data

Rangkaian penerima dan pengolah data terdiri dari Arduino Mega dan NRF24L01. Rangkaian ini akan menerima data yang sudah diakuisisi oleh rangkaian akuisisi data dan pengolahan data yang diklasifikasi menggunakan metode *K-Nearest Neighbor*. Jumlah pin NRF24L01 yang terhubung dengan Arduino Mega sama dengan Arduino Nano, namun konfigurasi pinnya berbeda. Pin GND pada NRF24L01 terhubung dengan pin GND pada Arduino Mega, pin VCC pada Arduino NRF24L01 terhubung dengan pin 3.3V pada Arduino Mega, pin CE terhubung dengan pin digital 7, pin CSN terhubung dengan pin 8, pin SCK terhubung dengan pin 52, pin MOSI terhubung dengan pin 51, dan pin MISO terhubung dengan pin 50. Arduino Mega ini akan terhubung dengan kabel USB ke laptop atau komputer untuk melihat output yang ditampilkan pada serial monitor Arduino IDE. Keterangan koneksi pin Arduino Mega dengan NRF24L01 terdapat pada tabel 5.4.

Tabel 5.4 Keterangan Pin NRF24L01 dengan Arduino Mega

Pin NRF24L01	Arduino Mega
MISO	50
MOSI	51
SCK	52
CE	7

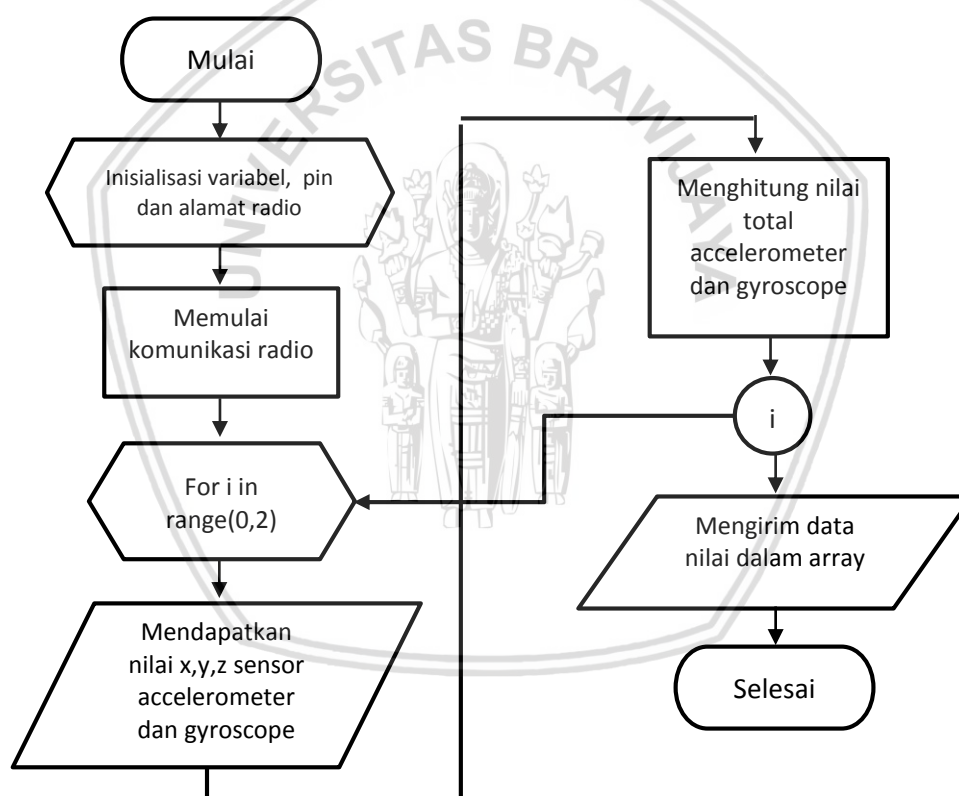
CSN	8
VCC	3V3
GND	GND

5.1.2 Perancangan Perangkat Lunak

Setelah dilakukan perancangan perangkat keras, perancangan perangkat lunak diperlukan untuk membuat logika sistem agar berjalan sesuai dengan tujuan. Perancangan perangkat lunak terdiri dari perangkat lunak untuk node sensor pengirim data dan perangkat lunak untuk penerima dan klasifikasi data dengan metode k-NN.

5.1.2.1 Perancangan Pengambilan Data Sensor

Perancangan perangkat lunak untuk node sensor pengirim data terdapat pada *flowchart* pada Gambar 5.3.



Gambar 5.3 Flowchart Pengambilan Data

Pada Gambar 5.3, diketahui bahwa pengambilan data dimulai dengan inisialisasi beberapa variabel yang digunakan, inisialisasi pin NRF24L01 yang terhubung dengan Arduino Nano dan inisialisasi alamat radio yang digunakan untuk mengirim data. Kemudian dilanjutkan dengan memulai komunikasi radio, sehingga proses pengiriman data bisa dilakukan. Pengambilan data berupa data mentah nilai x, y dan z sensor accelerometer dan gyroscope dilakukan tiga kali. Data mentah dari nilai sensor kemudian dilakukan proses penghitung nilai total

sensor accelerometer dan gyroscope. Setelah itu data-data yang telah diproses dijadikan data array untuk dikirimkan menggunakan modul komunikasi *wireless* ke alamat yang telah ditentukan. Rumus untuk menghitung total nilai sensor *accelerometer* terdapat pada Persamaan 5.1.

$$AT = \sqrt{Ax^2 + Ay^2 + Az^2} \quad (5.1)$$

Pada persamaan 5.1, AT (*Acceleration Total*) merupakan simbol untuk menyatakan total nilai sensor *accelerometer*. Ax merupakan variabel yang menyatakan komponen nilai sensor *accelerometer* terhadap sumbu x, Ay menyatakan komponen nilai *accelerometer* sumbu y dan Az menyatakan komponen nilai *accelerometer* sumbu z. Sedangkan rumus untuk menghitung total nilai sensor *gyroscope* terdapat pada Persamaan 5.2.

$$GT = \sqrt{Gx^2 + Gy^2 + Gz^2} \quad (5.2)$$

Pada persamaan 5.2, GT (*Gyroscope Total*) merupakan simbol untuk menyatakan total nilai sensor *gyroscope*. Gx merupakan variabel yang menyatakan komponen nilai sensor *gyroscope* terhadap sumbu x, Gy menyatakan komponen nilai *gyroscope* sumbu y dan Gz menyatakan komponen nilai *gyroscope* sumbu z.

Pada penelitian ini terdapat 2 macam sistem klasifikasi aktivitas manusia yang digunakan untuk pengujian. Sistem yang menggunakan satu sensor MPU6050 mempunyai 6 fitur untuk klasifikasinya, antara lain sebagai berikut:

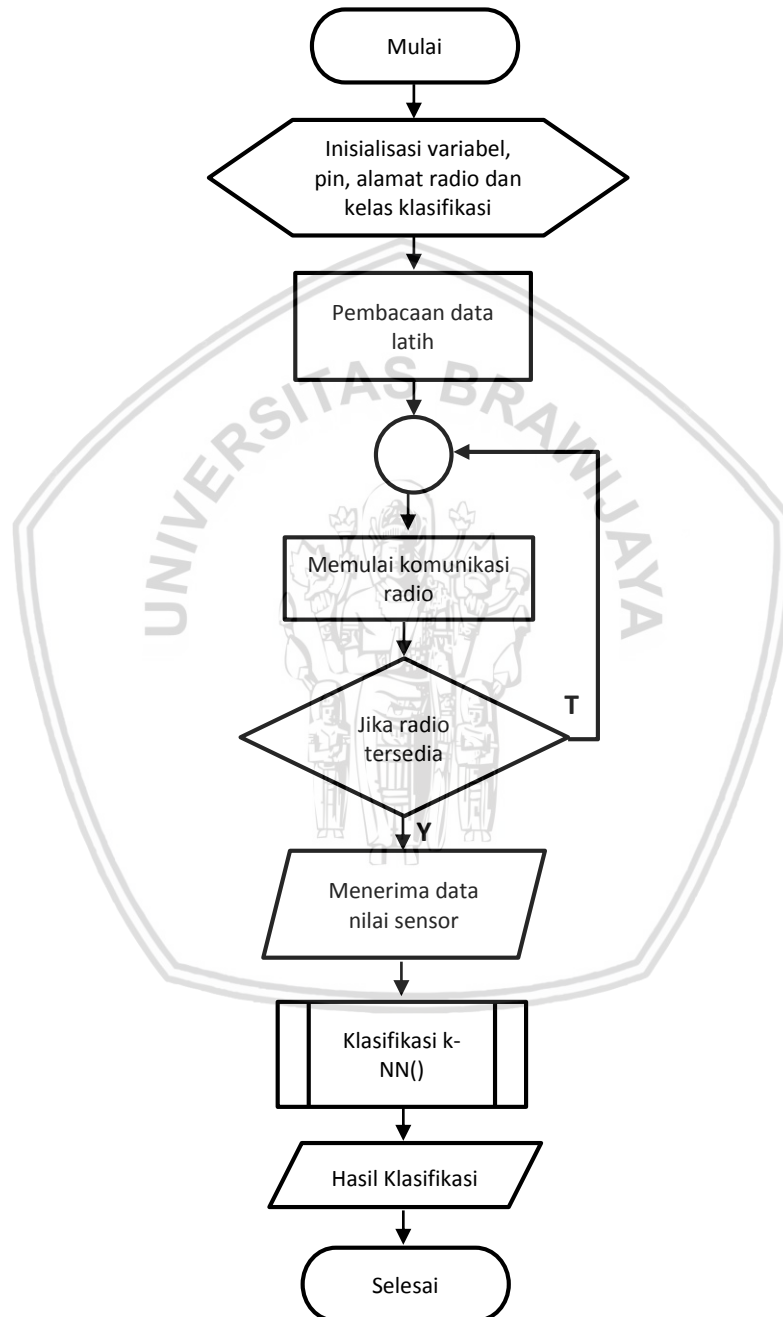
1. **Fitur 1:** AT pembacaan sensor pertama
2. **Fitur 2:** GT pembacaan sensor pertama
3. **Fitur 3:** AT pembacaan sensor kedua
4. **Fitur 4:** GT pembacaan sensor kedua
5. **Fitur 5:** AT pembacaan sensor ketiga
6. **Fitur 6:** GT pembacaan sensor ketiga

Sedangkan pada sistem yang menggunakan 2 sensor terdapat 12 fitur untuk klasifikasinya. 6 fitur pertama merupakan nilai sensor dari sensor pertama dan 6 fitur berikutnya merupakan nilai sensor dari sensor MPU6050 kedua. 12 fitur tersebut adalah sebagai berikut:

1. **Fitur 1:** AT pembacaan sensor pertama dari sensor MPU6050 pertama.
2. **Fitur 2:** GT pembacaan sensor pertama dari sensor MPU6050 pertama.
3. **Fitur 3:** AT pembacaan sensor kedua dari sensor MPU6050 pertama.
4. **Fitur 4:** GT pembacaan sensor kedua dari sensor MPU6050 pertama.
5. **Fitur 5:** AT pembacaan sensor ketiga dari sensor MPU6050 pertama.
6. **Fitur 6:** GT pembacaan sensor ketiga dari sensor MPU6050 pertama.
7. **Fitur 7:** AT pembacaan sensor pertama dari sensor MPU6050 kedua.
8. **Fitur 8:** GT pembacaan sensor pertama dari sensor MPU6050 kedua.
9. **Fitur 9:** AT pembacaan sensor kedua dari sensor MPU6050 kedua.
10. **Fitur 10:** GT pembacaan sensor kedua dari sensor MPU6050 kedua.
11. **Fitur 11:** AT pembacaan sensor ketiga dari sensor MPU6050 kedua.
12. **Fitur 12:** GT pembacaan sensor ketiga dari sensor MPU6050 kedua.

5.1.2.2 Perancangan Sistem Penerima Data Sensor

Data yang telah dikirim akan diterima oleh rangkaian Arduino Mega untuk dilakukan klasifikasi. Modul komunikasi wireless pada Arduino Mega juga mempunyai alamat yang sama seperti modul komunikasi yang terhubung dengan Arduino Nano. Flowchart atau diagram alir program penerima data dan klasifikasi ditunjukkan pada Gambar 5.4.



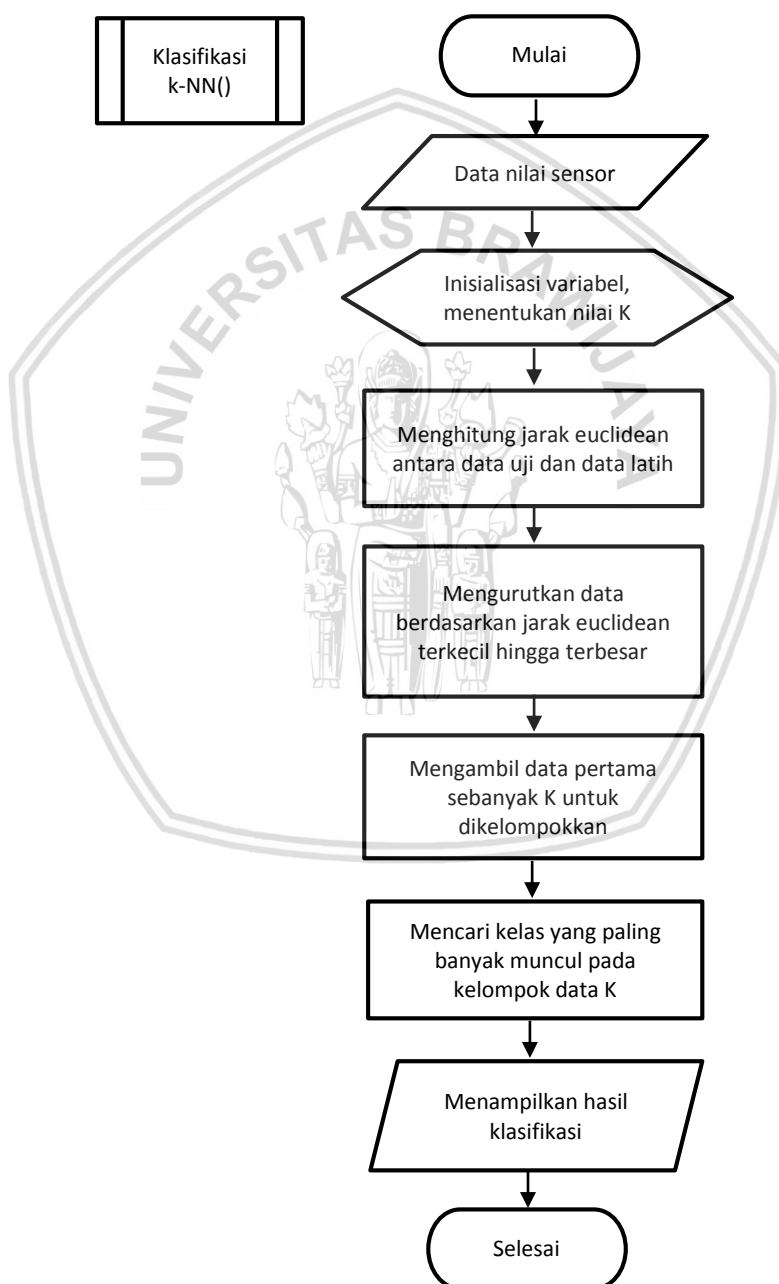
Gambar 5.4 Flowchart Penerima Data

Program dari perangkat lunak penerima data dimulai dengan inisialisasi pin, variabel dan alamat radio, kemudian inisialisasi kelas klasifikasi yang terdiri dari 4 kelas, yaitu berdiri, duduk, berbaring dan berjalan. Pembacaan data latih

digunakan sebagai acuan klasifikasi berdasarkan jarak terdekat. Setelah itu program memulai komunikasi radio untuk mendengarkan apakah terdapat perangkat yang mengirim data dengan alamat yang sama. Jika semua data sudah diterima, data tersebut akan diklasifikasi menggunakan algoritma *k-Nearest Neighbor*.

5.1.2.3 Perancangan Metode Klasifikasi *k-Nearest Neighbor*

Flowchart atau diagram alir metode *k-Nearest Neighbor* terdapat pada Gambar 5.5 tentang *flowchart* fungsi k-NN.



Gambar 5.5 *Flowchart* Fungsi k-NN()

Flowchart atau diagram alir 5.5 merupakan tahapan program metode klasifikasi k-Nearest Neighbor. Input dari program ini merupakan data yang sudah diterima oleh Arduino Mega yang berasal dari nilai sensor yang telah diolah oleh sistem pengirim data. Langkah awal dari metode ini adalah penentuan nilai k yang digunakan untuk menentukan jumlah tetangga terdekat yang akan dicari kelas yang paling banyak muncul. Kemudian dilakukan perhitungan jarak euclidean antara data uji dan seluruh data latih. Setelah itu data diurutkan berdasarkan jarak euclidean terkecil hingga terbesar dan data dari data pertama hingga data ke-k. Kelompok data k akan dicari kelas yang paling banyak muncul untuk dijadikan sebagai hasil klasifikasi.

Berikut merupakan contoh perhitungan manual metode k-Nearest Neighbor untuk klasifikasi aktivitas manusia. Contoh perhitungan ini menggunakan satu sensor yang diletakan di paha berdasarkan data latih pada LAMPIRAN dengan nilai data uji fitur1 = 26356, fitur2 = 1584, fitur3 = 26410, fitur4 = 1553, fitur5 = 25883 dan fitur6 = 11679. Langkah-langkah perhitungan manual metode k-Nearest Neighbor adalah sebagai berikut:

1. Menentukan nilai k atau jumlah tetangga terdekat. Misal pada perhitungan ini menggunakan nilai k=3.
2. Menghitung *Euclidean distance* (d) atau perhitungan jarak nilai antara data uji dengan masing-masing data latih yang telah diinisialisasi dengan rumus:

$$d_n = \sqrt{(f1_{latih} - f1_{uji})^2 + (f2_{latih} - f2_{uji})^2 \dots \dots (f6_{latih} - f6_{uji})^2} \quad (5.3)$$

Sehingga didapatkan hasil nilai *euclidean distance* dari 40 data latih adalah sebagai berikut:

Tabel 5.5 Hasil Perhitungan *Euclidean Distance*

No.	Jarak <i>euclidean</i>	Kelas
1	24076,17	Berdiri
2	10544,89	Berdiri
3	11410,84	Berdiri
4	15702,29	Berdiri
5	9444,62	Berdiri
6	21567,67	Berdiri
7	26184,48	Berdiri
8	21460,70	Berdiri
9	21810,64	Berdiri
10	20241,91	Berdiri
11	19130,73	Duduk
12	21443,26	Duduk
13	18601,81	Duduk
14	15869,44	Duduk
15	18856,96	Duduk
16	18518,04	Duduk

17	18670,17	Duduk
18	18418,75	Duduk
19	18986,62	Duduk
20	18385,78	Duduk
21	10278,11	Berbaring
22	18518,43	Berbaring
23	13258,47	Berbaring
24	18518,43	Berbaring
25	13384,71	Berbaring
26	13594,19	Berbaring
27	17604,63	Berbaring
28	13395,68	Berbaring
29	13326,48	Berbaring
30	18714,97	Berbaring
31	31457,32	Berjalan
32	17202,90	Berjalan
33	21954,20	Berjalan
34	26214,41	Berjalan
35	11540,57	Berjalan
36	17826,92	Berjalan
37	21594,15	Berjalan
38	14388,81	Berjalan
39	51555,72	Berjalan
40	25580,16	Berjalan

3. Setelah itu data hasil perhitungan *euclidean distance* diurutkan berdasarkan nilai yang paling kecil hingga terbesar. Hasil pengurutan data hasil perhitungan *euclidean distance* terdapat pada tabel 5.6.

Tabel 5.6 Hasil Sorting Nilai Euclidean

No.	Jarak <i>euclidean</i>	Kelas
1	9444,62	Berdiri
2	10278,11	Berbaring
3	10544,89	Berdiri
4	11410,84	Berdiri
5	11540,57	Berjalan
6	13258,47	Berbaring
7	13326,48	Berbaring
8	13384,71	Berbaring
9	13395,68	Berbaring
10	13594,19	Berbaring
11	14388,81	Berjalan
12	15702,29	Berdiri

13	15869,44	Duduk
14	17202,90	Berjalan
15	17604,63	Berbaring
16	17826,92	Berjalan
17	18385,78	Duduk
18	18418,75	Duduk
19	18518,04	Duduk
20	18518,43	Berbaring
21	18518,43	Berbaring
22	18601,81	Duduk
23	18670,17	Duduk
24	18714,97	Berbaring
25	18856,96	Duduk
26	18986,62	Duduk
27	19130,73	Duduk
28	20241,91	Berdiri
29	21443,26	Duduk
30	21460,70	Berdiri
31	21567,67	Berdiri
32	21594,15	Berjalan
33	21810,64	Berdiri
34	21954,20	Berjalan
35	24076,17	Berdiri
36	25580,16	Berjalan
37	26184,48	Berdiri
38	26214,41	Berjalan
39	31457,32	Berjalan
40	51555,72	Berjalan

4. Kemudian data yang telah diurutkan diambil data sebanyak k pertama untuk dicari kelas yang paling banyak muncul. Pada perhitungan ini digunakan nilai k sama dengan 3, sehingga data yang diambil sebanyak 3 data pertama, yang terdapat pada Tabel 5.7.

Tabel 5.7 Hasil Pengambilan 3 Data Pertama

No.	Jarak <i>euclidean</i>	Kelas
1	9444,62	Berdiri
2	10278,11	Berbaring
3	10544,89	Berdiri

5. Data yang telah diambil sebanyak k pertama, akan dicari kelas yang paling banyak muncul untuk dijadikan hasil klasifikasi. Pada tabel 5.7, diketahui bahwa kelas yang paling banyak muncul adalah kelas Berdiri yang muncul

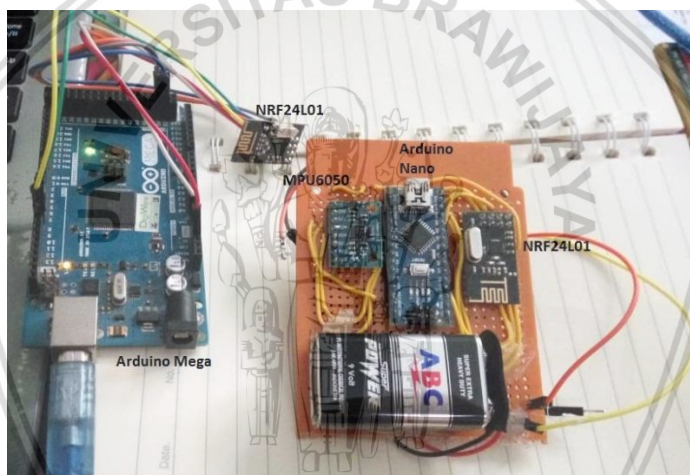
sebanyak 2 kali dan kelas Berbaring muncul sebanyak satu kali. Sehingga data uji tersebut masuk pada kelas **BERDIRI**.

5.2 Implementasi Sistem

Pada implementasi sistem, dilakukan realisasi pembuatan sistem baik dari sisi perangkat keras maupun perangkat lunak berdasarkan perancangan yang telah dilakukan. Selain implementasi perangkat keras dan perangkat lunak, pada subbab ini juga dijelaskan mengenai implementasi posisi peletakan sensor dan pengumpulan data latih.

5.2.1 Implementasi Perangkat Keras

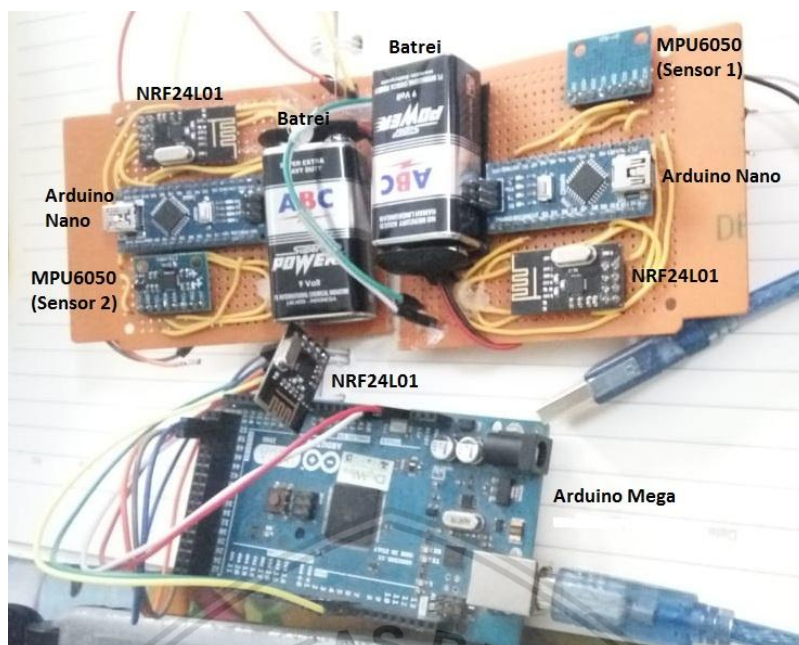
Tahap ini menjelaskan proses pengimplementasian perangkat keras berupa board Arduino Nano, NRF24L01 dan modul MPU6050 serta baterai yang akan menjadi daya dari arduino. Semua komponen dirangkai di PCB yang disolder sesuai dengan perancangan perangkat keras. Implementasi rangkaian perangkat keras yang menggunakan satu sensor ditunjukkan pada Gambar 5.6.



Gambar 5.6 Implementasi Perangkat Keras dengan Satu Sensor

Pada Gambar 5.6 terlihat bahwa sensor MPU6050 dirangkai dengan Arduino Nano dan NRF24L01 pada satu PCB yang akan digunakan untuk pengambilan data. Sedangkan Arduino Mega yang sudah dirangkai dengan modul NRF24L01 digunakan untuk menerima data dari Arduino Nano dan mengklasifikasikan data tersebut menggunakan metode *k-Nearest Neighbor*. Rangkaian Arduino Mega harus disambungkan dengan komputer/laptop secara serial menggunakan USB untuk melihat hasil klasifikasi yang akan ditampilkan di Serial Monitor Arduino IDE.

Selain sistem yang menggunakan satu sensor, penelitian ini juga melakukan pengujian menggunakan sistem dengan 2 sensor MPU6050. Rangkaian dan komponen yang digunakan sama dengan sistem yang menggunakan satu sensor, perbedaannya hanya pada jumlah rangkaian akuisisi data, dimana pada sistem yang menggunakan 2 sensor terdapat 2 rangkaian akuisisi data. Implementasi perangkat keras yang menggunakan 2 sensor terdapat pada Gambar 5.7.



Gambar 5.7 Implementasi Perangkat Keras dengan 2 Sensor

Pada Gambar 5.7, terdapat 2 rangkaian untuk pembacaan sensor. Kedua rangkaian tersebut mengirim hasil pembacaan sensor kepada Arduino Mega. Sehingga proses klasifikasi baru bisa dilakukan jika Arduino Mega sudah menerima semua data yang dikirimkan oleh sensor.

5.2.2 Implementasi Perangkat Lunak

Pada tahap implementasi perangkat lunak, dilakukan realisasi pembuatan program berdasarkan diagram alir yang telah dibuat pada perancangan sistem. Program untuk perangkat lunak sistem dibuat atau ditulis pada Arduino IDE. Sehingga lebih mudah dalam melakukan *compile* dan *upload* program ke *board* Arduino. Secara garis besar, terdapat 2 program utama dari sistem ini, yaitu program untuk akuisisi data sensor yang akan di-*upload* ke Arduino Nano dan program untuk menerima data dan klasifikasi yang akan di-*upload* ke Arduino Mega.

5.2.2.1 Implementasi Kode Program Pengambilan Data

Pada implementasi perangkat lunak, terdapat program untuk mengambil data sensor kemudian mengirimkannya serta program untuk menerima data sensor yang telah dikirim dan mengklasifikasikan. Untuk program pengambilan data dibutuhkan beberapa library untuk mendukung proses berjalannya program, diantaranya library RF24.h, SPI.h, Wire.h, I2Cdev.h dan MPU6050.h. Potongan kode program *import library* terdapat pada tabel 5.8.

Tabel 5.8 Kode Program Import Library

No	Kode Program
1	#include <nRF24L01.h>
2	#include <RF24.h>
3	#include <RF24_config.h>

4	#include <SPI.h>
5	#include "Wire.h"
6	#include "I2Cdev.h"
7	#include "MPU6050.h"

Sebelum menjalankan program utama, terdapat inisialisasi beberapa variabel yang dibutuhkan serta inisialisasi letak pin dan alamat radio. Potongan kode program untuk inisialisasi variabel, pin dan alamat radio terdapat pada tabel 5.9.

Tabel 5.9 Kode Program Inisialisasi Variabel

No	Kode Program
1	MPU6050 accelgyro;
2	int16_t ax, ay, az; // define accel as ax,ay,az
3	int16_t gx, gy, gz; // define gyro as gx,gy,gz
4	// Array Data yang akan dikirim
5	int data[6];
6	// pin CE , CS/N
7	RF24 radio(9, 10);
8	// alamat radio
9	const uint64_t pipe = 0xF0F0F0F0E1LL;

Pada tabel 5.9 terdapat inisialisasi variabel untuk menyimpan output nilai sensor accelerometer dan gyroscope yang berupa pada baris 2 dan 3. Kemudian terdapat deklarasi variabel array bernama data dengan panjang 6 elemen yang akan menyimpan hasil pengolahan nilai sensor yang terdapat pada baris ke-5. v NRF24L01 dan alamat radio yang digunakan untuk mengirim data.

Tabel 5.10 Kode Program Fungsi *Setup()*

No	Kode Program
1	Wire.begin(); // join I2C bus
2	accelgyro.initialize();
3	radio.begin();
4	radio.openWritingPipe(pipe);

Tabel 5.10 merupakan potongan kode program yang terdapat pada fungsi *setup()*. Pada baris ke-1 merupakan kode program yang digunakan untuk bergabung dengan jalur komunikasi I2C. Baris ke-2 digunakan untuk inisialisasi sensor MPU6050 yang akan dibaca datanya. Baris 3 dan 4 kode program untuk memulai komunikasi radio dengan NRF24L01 dan menyiapkan modul NRF24L01 untuk melakukan pengiriman data ke alamat yang sudah ditentukan.

Tabel 5.11 Kode Program Pembacaan Nilai Sensor

No	Kode Program
1	accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
2	data[0] = sqrt(abs(pow(ax, 2)) + abs(pow(ay, 2)) +
3	abs(pow(az, 2)));
4	data[1] = sqrt(abs(pow(gx, 2)) + abs(pow(gy, 2)) +
5	abs(pow(gz, 2)));
6	delay(500);
7	accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
8	data[2] = sqrt(abs(pow(ax, 2)) + abs(pow(ay, 2)) +
9	abs(pow(az, 2)));

10	data[3] = sqrt(abs(pow(gx, 2)) + abs(pow(gy, 2)) +
11	abs(pow(gz, 2)));
12	delay(500);
13	accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
14	data[4] = sqrt(abs(pow(ax, 2)) + abs(pow(ay, 2)) +
15	abs(pow(az, 2)));
16	data[5] = sqrt(abs(pow(gx, 2)) + abs(pow(gy, 2)) +
17	abs(pow(gz, 2)));
18	data[6] = 1;
19	

Pada tabel 5.11, baris pertama merupakan kode program untuk membaca nilai sensor yang disimpan di variabel-variabel yang telah ditentukan. Baris 2-3 merupakan perhitungan atau rumus dari total nilai *accelerometer*. Baris 4-5 merupakan perhitungan atau rumus dari total nilai sensor *gyroscope*. Pengambilan data sensor dilakukan 3 kali dengan jarak waktu setiap pengambilan selama 500 ms. Baris ke-19 berisi label mengenai data-data tersebut. Label 1 menyatakan bahwa data-data tersebut berasal dari sensor 1 dan label 2 menyatakan bahwa data-data tersebut berasal dari sensor 2. Label ini digunakan ketika implementasi sistem yang menggunakan 2 sensor. Hasil perhitungan total nilai sensor akan dimasukkan dalam array dengan nama data.

Tabel 5.12 Kode Program Pengiriman Data

No	Kode Program
1	radio.write(data, sizeof(data));
2	delay(5000);

Pada tabel 5.12 merupakan potongan program untuk mengirim data menggunakan komunikasi radio. Data yang dikirimkan dalam bentuk array. Data sensor dikirimkan kembali setelah delay selama 5000ms.

5.2.2.2 Implementasi Kode Program Penerima Data

Pada implementasi perangkat lunak, selain program untuk akuisisi data juga ada program untuk menerima data dan melakukan klasifikasi. Program tersebut akan di-*pload* pada Arduino Mega. Tabel 5.13 menunjukkan kode program *import library* untuk program penerima data.

Tabel 5.13 Kode Program Import Library Penerima Data

No	Kode Program
1	#include <nRF24L01.h>
2	#include <RF24.h>
3	#include <RF24_config.h>
4	#include <SPI.h>

Pada tabel 5.13 terdapat beberapa *library* yang digunakan untuk program penerima data, yaitu library nRF24L01.h, RF24.h, RF24_config.h dan SPI.h. *Library* tersebut digunakan untuk membantu proses pembuatan program. Sedangkan inisialisasi variabel-variabel yang digunakan oleh sistem yang menggunakan satu sensor terdapat pada tabel 5.14.

Tabel 5.14 Kode Program Inisialisasi Variabel Sistem dengan Satu Sensor

No	Kode Program
1	RF24 radio(7,8);
2	const uint64_t pipe = 0xF0F0F0F0E1LL;
3	int data[7];
4	int datauji[6];

Pada tabel 5.14, terdapat inisialisasi pin yang digunakan NRF24L01. Pada Arduino Mega, pin CE dan CSN NRF24L01 dihubungkan dengan pin 7 dan 8. Baris 2 menunjukkan inisialisasi alamat radio yang digunakan. Pada baris 3 terdapat inisialisasi variabel data yang digunakan untuk menerima data yang telah dikirimkan. Sedangkan pada baris 4 terdapat inisialisasi variabel array datauji dengan panjang array 6 elemen yang digunakan untuk menampung nilai datauji yang akan diklasifikasi. Sedangkan pada sistem yang menggunakan dua sensor, kode program inisialisasi variabel terdapat pada tabel 5.15.

Tabel 5.15 Kode Program Inisialisasi Variabel Sistem dengan 2 Sensor

No	Kode Program
1	RF24 radio(7,8);
2	const uint64_t pipe = 0xF0F0F0F0E1LL;
3	int data[7];
4	int datauji[12];

Pada Tabel 5.15 diketahui bahwa kode program inisialisasi untuk sistem yang menggunakan satu sensor tidak jauh berbeda dengan sistem yang menggunakan 2 sensor. Perbedaannya hanya pada kode program baris ke-4 saja, dimana variabel datauji mempunyai panjang elemen 12, yang menandakan bahwa sistem yang menggunakan 2 sensor mempunyai 12 fitur.

Tabel 5.16 Kode Program Fungsi *Setup()*

No	Kode Program
1	void setup() {
2	Serial.begin(9600);
3	radio.begin();
4	radio.openReadingPipe(1,pipe);
5	radio.startListening();
6	}

Tabel 5.16 merupakan kode program pada fungsi *setup()*. Fungsi *setup()* merupakan salah satu fungsi penting yang sudah disediakan oleh Arduino IDE. Baris ke-2 merupakan pengaturan untuk *baudrate Serial Monitor* atau kecepatan pengiriman karakter ke *Serial Monitor* Arduino IDE. Baris ke-3 digunakan untuk memulai komunikasi radio dan baris ke-4 dan ke-5 memberikan perintah bahwa sistem siap untuk menerima data melalui alamat yang sudah ditentukan.

Tabel 5.17 Kode Program Utama Penerima Data Sistem dengan Satu Sensor

No	Kode Program
1	void loop() {
2	if(radio.available()){
3	radio.read(data, sizeof(data));
4	datauji[0] = data[0];

5	datauji[1] = data[1];
6	datauji[2] = data[2];
7	datauji[3] = data[3];
8	datauji[4] = data[4];
9	datauji[5] = data[5];
10	Serial.print("Data Sensor : ");
11	Serial.print(datauji[0]);
12	Serial.print(", ");
13	Serial.print(datauji[1]);
14	Serial.print(", ");
15	Serial.print(datauji[2]);
16	Serial.print(", ");
17	Serial.print(datauji[3]);
18	Serial.print(", ");
19	Serial.print(datauji[4]);
20	Serial.print(", ");
21	Serial.println(datauji[5]);
22	size_t a = 0;
23	int berdiri, duduk, berbaring, berjalan = 0;
24	knn(datauji);
25	datauji[0] = datauji[1] = datauji[2] =
26	datauji[3] = datauji[4] = datauji[5] = 0;
27	}

Tabel 5.17 merupakan program utama dari program sistem yang menggunakan satu sensor yang terletak pada fungsi *loop()*. Baris ke-2 digunakan untuk mendeteksi apakah ada data yang dikirim. Jika terdapat data yang siap diterima, baris ke-3 menunjukkan sistem akan menerima data dengan memasukkan data yang diterima pada variabel data. Baris ke-4 sampai 9 digunakan untuk memindahkan data dari data yang diterima ke data uji. Baris ke-10 sampai 21 digunakan untuk menampilkan semua data sensor yang telah diterima. Baris ke-22 merupakan inisialisasi variabel a yang digunakan untuk keperluan klasifikasi. Baris ke-23 merupakan inisialisasi perhitungan kelas yang muncul. Baris ke-24 merupakan pemanggilan fungsi *knn()* dengan parameter data uji. Baris ke-25 sampai 26 digunakan untuk menghapus nilai yang tersimpan pada variabel data uji.

Program 5.17 merupakan program yang digunakan untuk menerima data dari sistem yang menggunakan satu sensor. Program utama dari kode program penerima data terdapat pada tabel 5.18.

Tabel 5.18 Kode Program Utama Penerima Data Sistem dengan 2 Sensor

No	Kode Program
1	if (radio.available()) {
2	a = 0;
3	berdiri, duduk, berbaring, berjalan = 0;
4	radio.read(data, sizeof(data));
5	if (data[6] == 1) {
6	datauji[0] = data[0];
7	datauji[1] = data[1];
8	datauji[2] = data[2];
9	datauji[3] = data[3];
10	datauji[4] = data[4];
11	datauji[5] = data[5];

12	} else if(data[6] == 2){
13	datauji[6] = data[0];
14	datauji[7] = data[1];
15	datauji[8] = data[2];
16	datauji[9] = data[3];
17	datauji[10] = data[4];
18	datauji[11] = data[5];
19	}
20	if(datauji[0] != 0 && datauji[6] != 0) {
21	knn(datauji);
22	datauji[0] = datauji[1] = datauji[2] =
23	datauji[3] = datauji[4] = datauji[5] = 0;
24	datauji[6] = datauji[7] = datauji[8] =
25	datauji[9] = datauji[10] = datauji[11] = 0;

Tabel 5.18 merupakan kode program utama yang digunakan untuk menerima data dari sistem yang menggunakan 2 sensor. Tahap awal penerimaan data sama dengan sistem yang menggunakan satu sensor, yaitu memeriksa apakah terdapat radio yang mengirimkan data. Jika ada sistem akan menerima data yang dimasukan pada variabel array data. Pada baris program nomor 5, isi dari variabel data indeks ke-6 dicek, jika nilainya 1 maka data yang diterima merupakan nilai dari sensor 1, sehingga data yang diterima disimpan pada variabel datauji indeks ke-0 hingga indeks ke-5. Jika tidak, nilai variabel data indeks ke-6 diperiksa lagi pada baris ke-12. Jika nilainya 2, maka data yang diterima merupakan nilai dari sensor 2, sehingga data disimpan pada variabel datauji indeks ke-6 hingga indeks ke-11. Pada baris program nomor 20, dilakukan pengecekan pada variabel datauji indeks ke-0 dan indeks ke 6 untuk memastikan bahwa semua data dari kedua sensor sudah diterima. Kemudian dilakukan pemanggilan fungsi knn() dengan parameter variabel array datauji. Setelah proses klasifikasi selesai, variabel array datauji nilainya dikosongkan.

5.2.2.3 Implementasi Kode Program Metode k-Nearest Neighbor

Implementasi kode program metode k-Nearest Neighbor digunakan untuk melakukan klasifikasi data sensor *accelerometer* dan *gyroscope* ke dalam beberapa kelas aktivitas manusia, yaitu berdiri, duduk, berbaring dan berjalan. Kode program metode k-Nearest Neighbor terdapat pada program penerima data yang di-*upload* pada Arduino Mega. Dalam pembuatan program, pertama melakukan inialisasi variabel yang digunakan. Kode program inialisasi terdapat pada Tabel 5.19.

Tabel 5.19 Kode Program Inialisasi Kelas Klasifikasi dan Variabel

No	Kode Program
1	String kelas[] = {"Berdiri",
2	"Duduk" ,
3	"Berbaring" ,
4	"Berjalan"
5	};
6	const size_t totalDataLatih = 40;
7	float hasilhitung[totalDataLatih][2],
8	hasilsort[totalDataLatih][2];
9	int k=3;

Pada Tabel 5.19, dilakuka inisialisasi kelas klasifikasi yang berjumlah 4 kelas yang terdiri dari kelas berdiri, duduk, berbaring dan berjalan. Inisialisasi kelas klasifikasi terdapat pada baris ke-1 sampai baris ke-4. Kemudian dilakukan inisialisasi jumlah data latih, yaitu sebanyak 40 data. Pada baris ke-7 hingga baris ke-8, terdapat deklarasi variabel yang digunakan untuk proses klasifikasi, yaitu variabel array hasilhitung yang akan menyimpan hasil perhitungan *euclidean distance* dan variabel array hasil sort yang akan menyimpan data setelah diurutkan berdasarkan jarak *euclidean* terkecil. Setelah itu terdapat inisialisasi nilai k yang digunakan pada program ini, yaitu nilai k sama dengan 3 yang terdapat pada baris ke-9. Untuk kode program data latih terdapat pada Tabel 5.20.

Tabel 5.20 Kode Program Pembacaan Data Latih dengan Satu Sensor

No	Kode Program
1	<code>int datalatih[totalDataLatih][7] =</code>
2	<code>{ {23443, 24764, 25177, 3227, 26567, 17071, 0},</code>
3	<code>{25150, 3339, 25143, 2089, 25030, 1479, 0},</code>
4	<code>{25235, 551, 25206, 1207, 25323, 454, 0},</code>
5	<code>{25847, 4145, 24835, 13252, 25572, 1664, 0},</code>
6	<code>{25212, 317, 25403, 3354, 25398, 2635, 0},</code>
7	<code>{15681, 609, 15675, 452, 15697, 275, 0},</code>
8	<code>{12859, 15024, 15561, 977, 15517, 1813, 0},</code>
9	<code>{15678, 453, 15753, 636, 15492, 594, 0},</code>
10	<code>{15438, 1973, 15341, 685, 15490, 496, 0},</code>
11	<code>{15490, 1144, 15535, 456, 15226, 4035, 0},</code>
12	<code>{16787, 6369, 18132, 1445, 17787, 842, 1},</code>
13	<code>{17235, 12343, 17542, 1424, 17667, 969, 1},</code>
14	<code>{17762, 468, 17728, 287, 17753, 372, 1},</code>
15	<code>{17766, 645, 17712, 410, 18947, 4461, 1},</code>
16	<code>{17573, 305, 17503, 388, 17531, 438, 1},</code>
17	<code>{17829, 665, 17710, 314, 17848, 384, 1},</code>
18	<code>{17774, 397, 17637, 303, 17663, 391, 1},</code>
19	<code>{17664, 1335, 17808, 693, 17699, 637, 1},</code>
20	<code>{17187, 655, 17432, 527, 17381, 651, 1},</code>
21	<code>{17827, 256, 17661, 891, 17557, 852, 1},</code>
22	<code>{22465, 724, 21106, 1200, 22267, 4720, 2},</code>
23	<code>{17460, 789, 17652, 255, 17882, 690, 2},</code>
24	<code>{22248, 1722, 22251, 561, 22162, 420, 2},</code>
25	<code>{17460, 789, 17652, 255, 17882, 690, 2},</code>
26	<code>{22154, 526, 22101, 436, 22224, 402, 2},</code>
27	<code>{22070, 342, 22089, 407, 22131, 241, 2},</code>
28	<code>{17874, 264, 17500, 354, 17598, 2364, 2},</code>
29	<code>{22187, 494, 22093, 381, 22113, 425, 2},</code>
30	<code>{22131, 455, 22231, 295, 22247, 445, 2},</code>
31	<code>{17725, 245, 17745, 526, 17638, 284, 2},</code>
32	<code>{25672, 31681, 25038, 1688, 24961, 2706, 3},</code>
33	<code>{25054, 526, 25496, 7776, 24452, 14410, 3},</code>
34	<code>{19536, 10805, 16857, 14559, 17523, 16167, 3},</code>
35	<code>{26302, 23102, 25130, 13305, 25915, 2491, 3},</code>
36	<code>{25158, 9746, 24448, 6461, 24913, 5658, 3},</code>
37	<code>{25836, 11695, 25676, 14140, 25890, 4174, 3},</code>
38	<code>{18480, 10413, 16185, 8232, 16002, 2761, 3},</code>
39	<code>{25333, 3073, 25607, 15145, 26307, 15946, 3},</code>
40	<code>{32584, 26715, 24515, 5743, 24581, -32647, 3},</code>
41	<code>{24258, 6711, 26034, 21406, 25832, 26824, 3}</code>
42	<code>};</code>

Tabel 5.20 merupakan kode program untuk memasukkan data latihan ke program. Variabel `datalatih` berbentuk array 2 dimensi dengan total baris 40 data. Setiap baris data mempunyai 7 elemen yang terdiri dari 6 fitur nilai sensor dan 1 nilai yang digunakan untuk label data latihan. Label 0 menandakan bahwa data latihan tersebut termasuk kelas Berdiri, label 1 menandakan kelas Duduk, label 2 menandakan kelas Berbaring dan label 3 menandakan kelas Berjalan. Masing-masing kelas mempunyai 10 data latihan, sehingga total data latihan yang digunakan sebanyak 40 data latihan. Program tersebut digunakan untuk memasukkan data latihan dari sistem yang menggunakan satu sensor. Sementara program untuk memasukkan data latihan yang digunakan sistem dengan 2 sensor terdapat pada Tabel 5.21.

Tabel 5.21 Kode Program Pembacaan Data Latihan dengan 2 Sensor

No	Kode Program
1	<code>int datalatih[totalDataLatih][13] =</code>
2	<code>{ {24865, 1865, 24845, 7290, 24693, 2396, 15464,</code>
3	<code>4819, 15357, 4871, 15514, 2790, 0},</code>
4	<code>{24894, 897, 25028, 1696, 24449, 12727, 15999,</code>
5	<code>4354, 16796, 4420, 15949, 236, 0},</code>
6	<code>{26217, 21895, 26261, 1306, 25123, 3404,</code>
7	<code>15202, 10595, 14537, 6892, 17115, 15990, 0},</code>
8	<code>{25000, 1562, 24777, 2467, 24814, 775, 15614,</code>
9	<code>2270, 15518, 400, 15548, 312, 0},</code>
10	<code>{24844, 783, 24735, 333, 24828, 471, 15541,</code>
11	<code>1818, 15406, 291, 15413, 868, 0},</code>
12	<code>{23729, 27821, 24195, 12673, 24480, 6558,</code>
13	<code>15331, 4457, 15750, 975, 15776, 2553, 0},</code>
14	<code>{23470, 19814, 24680, 3432, 25087, 2087,</code>
15	<code>16395, 2263, 18430, 2958, 15693, 8607, 0},</code>
16	<code>{24832, 6718, 25096, 10289, 24538, 2953,</code>
17	<code>15832, 18344, 15726, 21735, 15798, 4810, 0},</code>
18	<code>{24680, 743, 24732, 558, 24720, 518, 15407,</code>
19	<code>9196, 15368, 1468, 15420, 414, 0},</code>
20	<code>{24706, 2664, 24899, 1662, 24821, 4368, 15140,</code>
21	<code>4725, 15171, 2590, 15065, 4839, 0},</code>
22	<code>{25036, 1687, 24581, 3297, 24750, 2246, 15108,</code>
23	<code>1529, 15889, 5644, 15026, 7969, 1},</code>
24	<code>{24422, 705, 24837, 357, 24576, 174, 15772,</code>
25	<code>5143, 15202, 1546, 15617, 621, 1},</code>
26	<code>{24953, 454, 24786, 637, 24866, 545, 15244,</code>
27	<code>309, 15343, 665, 15596, 659, 1},</code>
28	<code>{24775, 478, 24824, 530, 25019, 123, 15857,</code>
29	<code>649, 15452, 729, 15511, 642, 1},</code>
30	<code>{25083, 192, 24859, 518, 25009, 296, 15335,</code>
31	<code>423, 15321, 1091, 15465, 874, 1},</code>
32	<code>{24664, 403, 24906, 340, 24829, 509, 15618,</code>
33	<code>603, 15565, 1184, 15294, 1079, 1},</code>
34	<code>{26100, 359, 26267, 369, 26164, 480, 17012,</code>
35	<code>301, 17096, 444, 17240, 377, 1},</code>
36	<code>{26095, 447, 26155, 445, 25990, 504, 17198,</code>
37	<code>389, 17071, 481, 17321, 359, 1},</code>
38	<code>{26243, 11651, 24183, 25963, 24416, 11441,</code>
39	<code>16764, 739, 17958, 397, 17156, 794, 1},</code>
40	<code>{25151, 3048, 25255, 4441, 25003, 1404, 16954,</code>
41	<code>372, 17269, 440, 17195, 431, 1},</code>

42	{25168, 4012, 24536, 2982, 25506, 882, 17693,
43	1244, 17939, 1074, 17214, 764, 2},
44	{22558, 421, 22415, 500, 22554, 507, 17730,
45	246, 17885, 263, 17722, 221, 2},
46	{23225, 2129, 23095, 494, 23056, 2449, 22080,
47	23885, 19160, 29224, 17870, 5920, 2},
48	{22890, 362, 22658, 319, 22948, 634, 17776,
49	467, 18262, 831, 17582, 671, 2},
50	{22372, 520, 22687, 4284, 22395, 2648, 17483,
51	1381, 18373, 3754, 17855, 1338, 2},
52	{22960, 685, 22803, 801, 22709, 397, 17564,
53	1201, 17251, 388, 17918, 353, 2},
54	{22784, 434, 22800, 432, 22733, 422, 17737,
55	326, 17654, 249, 17405, 500, 2},
56	{22689, 544, 22511, 520, 22455, 540, 17842,
57	258, 17761, 300, 17862, 228, 2},
58	{22513, 387, 22571, 433, 22567, 414, 17764,
59	254, 17919, 248, 17664, 221, 2},
60	{22460, 558, 22597, 307, 22522, 497, 17774,
61	306, 17770, 246, 17907, 395, 2},
62	{27230, 3787, 24313, 21754, 23060, 6121,
63	14538, 9389, 17554, 18552, 14655, 7878, 3},
64	{26075, 14281, 24614, 10882, 28093, 30891,
65	15198, 19375, 16313, 4608, 25630, -26702, 3},
66	{27592, -32420, 26753, 3999, 24891, 13283,
67	22423, -31528, 14632, 20714, 12911, 30584, 3},
68	{25402, -29294, 25123, 3104, 22685, 26560,
69	17829, -23406, 16868, -18604, 17269, -29203, 3},
70	{24819, 20472, 27420, 15823, 26330, 2340,
71	16344, 6570, 16539, 4905, 15751, 9370, 3},
72	{25757, 9838, 25368, 4337, 23782, 30404,
73	15592, 6622, -25496, 9954, 15316, 3},
74	{27783, 24733, 25884, -30326, 22499, 5727,
75	10434, -24330, 16841, 19996, -32170, -14315, 3},
76	{21748, -26041, 25975, 5702, 22855, 28317,
77	12110, 28048, 12596, 18527, 11289, 31425, 3},
78	{23235, -17855, 30362, 17846, 27646, -26178,
79	15616, 28176, 13675, 20271, 17975, -27760, 3},
80	{24492, 16580, 30490, -17088, 23848, -28049,
81	15616, 28176, 13675, 20271, 17975, -27760, 3}
82	};

Tabel 5.21 merupakan kode program untuk memasukkan data latih pada program. Pada program untuk sistem yang menggunakan 2 sensor, setiap data latih mempunyai 13 elemen nilai. 6 elemen pertama merupakan fitur dari sensor pertama, 6 elemen kedua merupakan fitur dari sensor kedua dan elemen ke-13 merupakan label dari data latih yang menunjukkan kelas.

Tabel 5.22 Kode Program Perhitungan *Euclidean Distance* dengan Satu Sensor

No	Kode Program
1	void knn (int du[6]) {
2	for (size_t i = 0; i < totalDataLatih; i++){
3	hasilhitung[i][0] = sqrt(abs(pow(datalatih[i][0], 2)-
4	pow(du[0], 2))+abs(pow(datalatih[i][1], 2)-pow(du[1], 2))
5	+ abs(pow(datalatih[i][2], 2)-pow(du[2], 2)) +
6	abs(pow(datalatih[i][3], 2)-pow(du[3], 2))
7	}

8	+ abs(pow(datalatih[i][4], 2)-
9	pow(du[4], 2)) + abs(pow(datalatih[i][5], 2)-pow(du[5],
10	2)));
11	hasilhitung[i][1] = datalatih[i][6];
12	}

Tabel 5.22 menunjukkan kode program yang digunakan untuk menghitung nilai *euclidean* sistem dengan satu sensor dari data uji dengan masing-masing data latih. Pada baris pertama, parameter dari fungsi knn() merupakan data array dengan panjang 6 elemen, yaitu data uji. Pada proses perhitungan *euclidean distance*, hasil perhitungan dimasukan ke dalam variabel array hasilhitung indeks ke-0 dan indeks ke-1 merupakan label kelas dari data tersebut. Sedangkan kode program perhitungan euclidean distance untuk sistem dengan 2 sensor terdapat pada Tabel 5.23.

Tabel 5.23 Kode Program Perhitungan *Euclidean Distance* dengan 2 Sensor

No	Kode Program
1	void knn (int du[12]) {
2	for (size_t i = 0; i < totalDataLatih; i++){
3	hasilhitung[i][0] = sqrt(abs(pow(datalatih[i][0], 2)-
4	pow(du[0], 2))+abs(pow(datalatih[i][1], 2)-pow(du[1], 2))
5	+ abs(pow(datalatih[i][2], 2)-
6	pow(du[2], 2)) + abs(pow(datalatih[i][3], 2)-pow(du[3],
7	2))
8	+ abs(pow(datalatih[i][4], 2)-
9	pow(du[4], 2)) + abs(pow(datalatih[i][5], 2)-pow(du[5],
10	2))
11	+ abs(pow(datalatih[i][6], 2)-
12	pow(du[6], 2)) + abs(pow(datalatih[i][7], 2)-pow(du[7],
13	2))
14	+ abs(pow(datalatih[i][8], 2)-
15	pow(du[8], 2)) + abs(pow(datalatih[i][9], 2)-pow(du[9],
16	2))
17	+ abs(pow(datalatih[i][10], 2)-
18	pow(du[10], 2)) + abs(pow(datalatih[i][11], 2)-pow(du[11],
19	2)));
20	hasilhitung[i][1] = datalatih[i][12];
21	}

Tabel 5.23 merupakan kode program perhitungan *euclidean distance* yang digunakan untuk sistem yang menggunakan 2 sensor. Perbedaan program ini dengan program perhitungan euclidean distance untuk sistem yang menggunakan satu sensor adalah pada jumlah fitur yang digunakan. Sistem dengan 2 sensor mempunyai 12 fitur, sehingga parameter fungsi knn() yang digunakan merupakan variabel array dengan panjang elemen 12.

Tabel 5.24 Kode Program *Sorting Data*

No	Kode Program
1	memcpy(hasilsort, hasilhitung, sizeof(hasilsort));
2	for (size_t i = 1; i < totalDataLatih; i++) {
3	for (size_t j = i; j > 0 && hasilsort[j-1][0] >
4	hasilsort[j][0]; j--) {
5	float tmp[2];
6	memcpy(tmp, hasilsort[j-1], sizeof(tmp));

7	memcpy(hasilsort[j-1], hasilsort[j],
8	sizeof(hasilsort[j-1]));
9	memcpy(hasilsort[j], tmp, sizeof(hasilsort[j]));
10	}
11	}

Tabel 5.12 merupakan kode program untuk mengurutkan data hasil proses perhitungan euclidean distance. Data diurutkan berdasarkan jarak euclidean terkecil hingga terbesar. Data yang telah diurutkan disimpan pada variabel array hasilsort.

Tabel 5.25 Kode Program Voting Jumlah Kelas yang Muncul

No	Kode Program
1	while (a < k) {
2	Serial.println();
3	Serial.print("k");
4	Serial.print(" = ");
5	Serial.print(a+1);
6	Serial.print(" -> ");
7	Serial.print(hasilsort[a][0]);
8	Serial.print(" -> ");
9	Serial.print(kelas[int(hasilsort[a][1])]);
10	
11	if(hasilsort[a][1] == 0) {
12	berdiri++;
13	} else if(hasilsort[a][1] == 1) {
14	duduk++;
15	} else if(hasilsort[a][1] == 2) {
16	berbaring++;
17	} else if(hasilsort[a][1] == 3) {
18	berjalan++;
19	}
20	a++;
21	}
22	Serial.println();
23	Serial.print("berdiri = ");
24	Serial.print(berdiri);
25	Serial.println();
26	Serial.print("duduk = ");
27	Serial.print(duduk);
28	Serial.println();
29	Serial.print("berbaring = ");
30	Serial.print(berbaring);
31	Serial.println();
32	Serial.print("berjalan = ");
33	Serial.print(berjalan);
34	Serial.println();

Tabel 5.25 merupakan kode program untuk menghitung jumlah kelas yang muncul pada k data pertama. Pada baris pertama, terdapat perulangan sebanyak k yang digunakan untuk menampilkan data pertama hingga data ke-k dari data yang telah diurutkan. Baris ke-2 sampai baris ke-9 merupakan kode program untuk menampilkan hasil perhitungan *euclidean distance* dari data pertama hingga data ke-k. Baris ke-11 hingga baris ke-18 merupakan kode program yang digunakan

untuk *voting* kelas yang muncul. Jika data latih yang termasuk dalam k data pertama akan dicek apakah termasuk dalam kelas berdiri, duduk, berbaring atau berjalan. Jika masuk ke dalam salah satu kelas tersebut, nilai variabel berdiri, duduk, berbaring dan berjalan akan bertambah satu. Kemudian hasil voting terhadap kelas klasifikasi akan ditampilkan. Baris ke-23 sampai baris ke-33 merupakan kode program yang digunakan untuk menampilkan hasil voting.

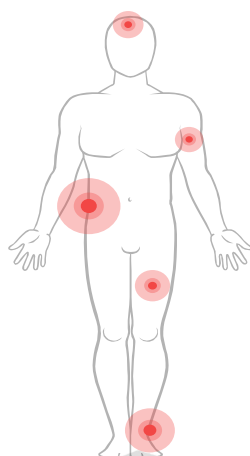
Tabel 5.26 Kode Program Penentuan Hasil Klasifikasi

No	Kode Program
1	if(berdiri>duduk && berdiri>berbaring && berdiri>berjalan){
2	Serial.println("Hasil klasifikasi adalah
3	BERDIRI");
4	} else if(duduk>berdiri && duduk>berbaring &&
5	duduk>berjalan){
6	Serial.println("Hasil klasifikasi adalah DUDUK");
7	} else if(berbaring>berdiri && berbaring>duduk &&
8	berbaring>berjalan){
9	Serial.println("Hasil klasifikasi adalah
10	BERBARING");
11	} else if(berjalan>berdiri && berjalan>duduk &&
12	berjalan>berbaring){
13	Serial.println("Hasil klasifikasi adalah
14	BERJALAN");
15	} else {
16	Serial.print("Hasil klasifikasi adalah ");
17	Serial.println(kelas[int(hasilsort[0][1])]);
18	}

Tabel 5.26 merupakan kode program yang digunakan untuk menentukan hasil klasifikasi. Hasil *voting* yang telah dilakukan pada variabel berdiri, duduk, berbaring dan berjalan, akan dilakukan perbandingan untuk menentukan kelas yang paling banyak muncul. Masing-masing variabel mewakili kelas sesuai dengan namanya. Pada baris ke-1, dilakukan pengecekan jika kelas berdiri lebih banyak muncul daripada kelas yang lain, maka hasil klasifikasi adalah berdiri. Pada baris ke-4 dilakukan pengecekan jika kelas duduk lebih banyak muncul daripada kelas yang lain, maka hasil klasifikasi adalah duduk. Pengecekan juga dilakukan pada variabel berbaring yang terdapat pada baris ke-7 dan variabel berjalan yang terdapat pada baris ke-11. Jika ada lebih dari satu kelas dengan *voting* terbanyak, maka hasil klasifikasi adalah kelas yang mempunyai jarak terdekat dengan data uji.

5.2.3 Implementasi Penentuan Posisi Peletakan Sensor

Pada implementasi penentuan posisi peletakan sensor, terdapat penentuan peletakan sensor pada beberapa bagian tubuh untuk dicari posisi peletakan yang menghasilkan akurasi paling baik. Penelitian ini menggunakan 5 anggota tubuh yang berbeda untuk peletakan sensor, yaitu kepala, lengan, pinggang, paha dan kaki bagian bawah. Gambar peletakan sensor ditunjukkan pada Gambar 5.8 tentang posisi peletakan sensor pada bagian tubuh.



Gambar 5.8 Posisi Peletakan Sensor pada Bagian Tubuh

Pada sistem yang menggunakan satu sensor, posisi peletakan sensor yang digunakan sesuai dengan Gambar 5.8 terdapat pada Tabel 5.27.

Tabel 5.27 Posisi Peletakan Satu Sensor

POSISI KE-	LETAK SENSOR
Posisi 1	Paha
Posisi 2	Kaki Bagian Bawah
Posisi 3	Lengan
Posisi 4	Pinggang
Posisi 5	Kepala

Sedangkan pada sistem yang menggunakan dua sensor, terdapat kombinasi dari anggota tubuh yang menjadi posisi peletakan sensor. Dari 5 posisi peletakan sensor pada Gambar 5.8, dibuat kombinasi 2 posisi setiap pengujian. Perhitungan kombinasi peletakan sensor yang memungkinkan terjadi dapat dihitung menggunakan rumus pada Persamaan 5.4.

$$C_{(n,r)} = \frac{n!}{r!(n-r)!} \quad (5.4)$$

Dari 5 posisi peletakan sensor, dibuat kombinasi setiap 2 posisi, sehingga didapatkan jumlah kombinasi sebagai berikut:

$$C_{(5,2)} = \frac{5!}{2!(5-2)!}$$

$$C_{(5,2)} = \frac{5!}{2! \times 3!}$$

$$C_{(5,2)} = \frac{5 \times 4 \times 3 \times 2 \times 1}{2 \times 1 \times 3 \times 2 \times 1}$$

$$C_{(5,2)} = 10$$

Berdasarkan perhitungan kombinasi 2 peletakan sensor terhadap 5 posisi bagian tubuh, didapatkan 10 kombinasi peletakan sensor yang akan dilakukan pengujian untuk mendapatkan posisi dengan akurasi paling tinggi. Kombinasi peletakan sensor untuk sistem yang menggunakan sensor terdapat pada Tabel 5.28.

Tabel 5.28 Kombinasi Posisi Peletakan 2 Sensor

POSISI KE-	LETAK SENSOR
Posisi 1	Paha dan kaki bagian bawah
Posisi 2	Paha dan lengan
Posisi 3	Paha dan pinggang
Posisi 4	Paha dan kepala
Posisi 5	Kaki bagian bawah dan lengan
Posisi 6	Kaki bagian bawah dan pinggang
Posisi 7	Kaki bagian bawah dan kepala
Posisi 8	Lengan dan pinggang
Posisi 9	Lengan dan kepala
Posisi 10	Pinggang dan kepala

5.2.4 Implementasi Pengambilan Data Latih

Pada metode klasifikasi k-Nearest Neighbor, data latih sangat diperlukan karena untuk menentukan hasil klasifikasi, dibutuhkan perhitungan yang melibatkan nilai dari data latih. Pada penelitian ini, data latih didapatkan dari percobaan langsung dengan meletakkan sensor sesuai dengan implementasi peletakan sensor. Kemudian melakukan aktivitas sederhana yang telah ditentukan, yaitu berdiri, duduk, berbaring dan berjalan. Nilai sensor yang didapatkan pada proses tersebut digunakan sebagai data latih. Untuk bisa mendapatkan nilai sensor sebagai data latih, Arduino Mega yang menjadi penerima data diberikan kode program seperti pada Tabel 5.29.

Tabel 5.29 Kode Program Mengambil Data Latih

No	Kode Program
1	void loop(void){
2	if(radio.available()){
3	radio.read(data, sizeof(data));
4	Serial.print("Data Sensor ");
5	Serial.print(data[6]);
6	Serial.print(" = ");
7	Serial.print("{");
8	Serial.print(data[0]);
9	Serial.print(", ");
10	Serial.print(data[1]);
11	Serial.print(", ");

12	Serial.print(data[2]);
13	Serial.print(", ");
14	Serial.print(data[3]);
15	Serial.print(", ");
16	Serial.print(data[4]);
17	Serial.print(", ");
18	Serial.print(data[5]);
19	Serial.println("},");
20	}
21	}

Tabel 5.29 menunjukkan kode program yang digunakan untuk pengambilan data latih. Kode program tersebut berupa pada fungsi loop() yang merupakan fungsi utama pada Arduino IDE. Program yang terdapat pada fungsi loop() akan dijalankan berulang-ulang. Pada baris ke-2 dilakukan pengecekan apakah ada modul radio yang akan mengirimkan data ke alamat yang sama dengan program tersebut. Jika ada, sistem akan menerima data dengan kode program pada baris ke-3. Pada baris ke-5 dilakukan identifikasi nilai sensor, untuk mengetahui nilai sensor yang diterima berasal dari sensor pertama atau sensor kedua.

Setiap posisi peletakan sensor seperti yang telah dijelaskan pada subbab implementasi posisi peletakan sensor, diambil data latih sebanyak 40 data. Dari 40 data tersebut, 10 data berasal dari kelas berdiri, 10 data dari kelas duduk, 10 data dari kelas berbaring dan 10 data dari kelas berjalan. Kemudian 40 data latih yang telah diambil dimasukkan ke dalam kode program metode *k-Nearest Neighbor*. Seluruh data latih yang digunakan, baik pada sistem yang menggunakan satu sensor maupun pada sistem yang menggunakan dua sensor, terdapat pada LAMPIRAN.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan pengujian sistem serta analisis hasil pengujian. Tujuan dari pengujian adalah untuk mengetahui hasil implementasi sistem sudah sesuai dengan yang diharapkan atau belum. Ada beberapa pengujian yang dilakukan, yaitu pengujian fungsional sistem, pengujian nilai k, pengujian akurasi sistem dan pengujian waktu komputasi metode *k-Nearest Neighbor*. Pengujian fungsional sistem bertujuan untuk mengetahui kinerja fungsi-fungsi utama sistem. Pengujian ini terdiri dari pengujian pembacaan nilai sensor, pengujian pengiriman data sensor dan pengujian tampilan pada *Serial Monitor* Arduino IDE. Setiap pengujian terdapat tujuan pengujian, peralatan dan prosedur pengujian. Kemudian hasil dan analisis hasil pengujian juga ditampilkan pada bab ini. Berikut dijelaskan skenario dan hasil pengujian pada masing-masing subbab.

6.1 Pengujian Fungsional Sistem

Pengujian ini dilakukan untuk menguji fungsional sistem sesuai dengan kebutuhan fungsional sistem yang telah dijelaskan pada Bab 4. Pengujian fungsional terdiri dari pengujian pembacaan nilai sensor, pengujian pengiriman data sensor dan pengujian tampilan pada *Serial Monitor* Arduino IDE.

6.1.1 Pengujian Pembacaan Nilai Sensor MPU6050

Sensor MPU6050 merupakan sensor *6-axis* yang terdiri dari *3-axis* sensor *accelerometer* dan *3-axis* sensor *gyroscope*. Pengujian sensor ini dilakukan untuk menguji sensor bisa membaca nilai sensor *accelerometer* dan *gyroscope* serta mengolahnya menjadi total nilai *accelerometer* dan total nilai *gyroscope*.

6.1.1.1 Tujuan Pengujian

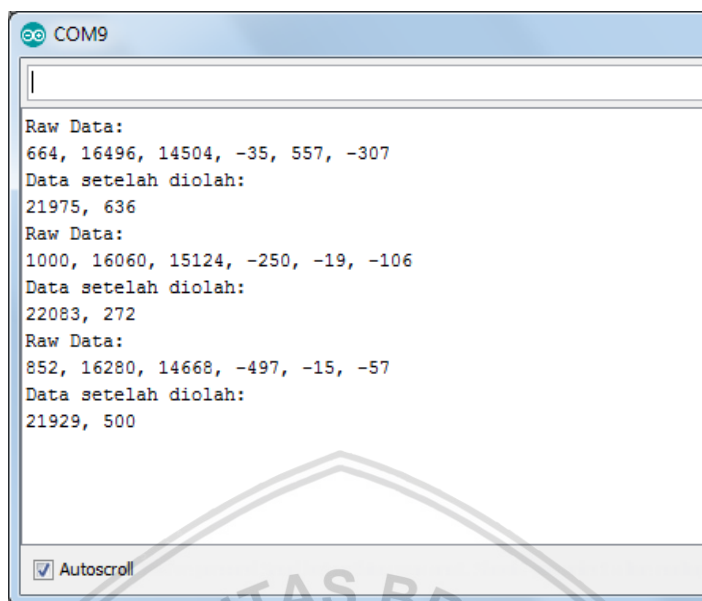
Pengujian pembacaan nilai sensor MPU6050 adalah untuk mengetahui apakah sensor tersebut bisa membaca data mentah atau *raw data* sensor *accelerometer* dan *gyroscope*. Serta mengolahnya menjadi total nilai *accelerometer* dan total nilai *gyroscope*.

6.1.1.2 Prosedur Pengujian

Berikut prosedur yang dilakukan untuk melakukan pengujian pembacaan nilai sensor MPU6050:

1. Menghubungkan mikrokontroler Arduino Nano yang sudah dirangkai dengan sensor MU6050 ke komputer/laptop.
2. Membuka aplikasi Arduino IDE dan mengatur *port*-nya.
3. Meng-*upload* kode program pembacaan nilai sensor ke Arduino Nano. Jika proses upload berhasil, LED pada Arduino Nano akan menyala dan pada Arduino IDE akan muncul keterangan "*done uploading*".
4. Membuka *Serial Monitor* Arduino IDE.
5. Mengamati hasil pembacaan sensor pada *Serial Monitor* Arduino IDE.

6.1.1.3 Hasil dan Analisis Pengujian



Gambar 6.1 Hasil Pengujian Pembacaan Nilai Sensor

Pada Gambar 6.1 diketahui bahwa sistem bisa melakukan pembacaan sensor *accelerometer* dan *gyroscope*. Sistem bisa membaca data mentah atau *raw data* dari nilai sensor. Kemudian bisa melakukan pengolahan data mentah sensor menjadi total nilai *accelerometer* dan total nilai *gyroscope*. Berdasarkan hasil pengujian, dapat dikatakan bahwa pengujian pembacaan data sensor berhasil.

6.1.2 Pengujian Pengiriman Data Sensor

Pada sistem ini, data sensor yang telah didapatkan dikirim dari Arduino Nano ke Arduino Mega menggunakan modul komunikasi wireless NRF24L01. Modul NRF24L01 bekerja pada frekuensi radio 2,4 GHz. Pengujian ini dilakukan dengan mengirimkan data sensor yang telah dibaca dari Arduino Nano. Sensor yang digunakan pada pengujian ini sebanyak dua sensor, sehingga Arduino Mega akan menerima dua data dari pengirim yang berbeda.

6.1.2.1 Tujuan Pengujian

Pengujian pengiriman data sensor bertujuan untuk melihat hasil pengiriman dan penerimaan data sensor berhasil dilakukan atau belum. Data yang dikirim berasal dari 2 sensor yang berbeda. Pengujian ini juga bertujuan untuk mengetahui apakah ada data yang hilang atau tidak.

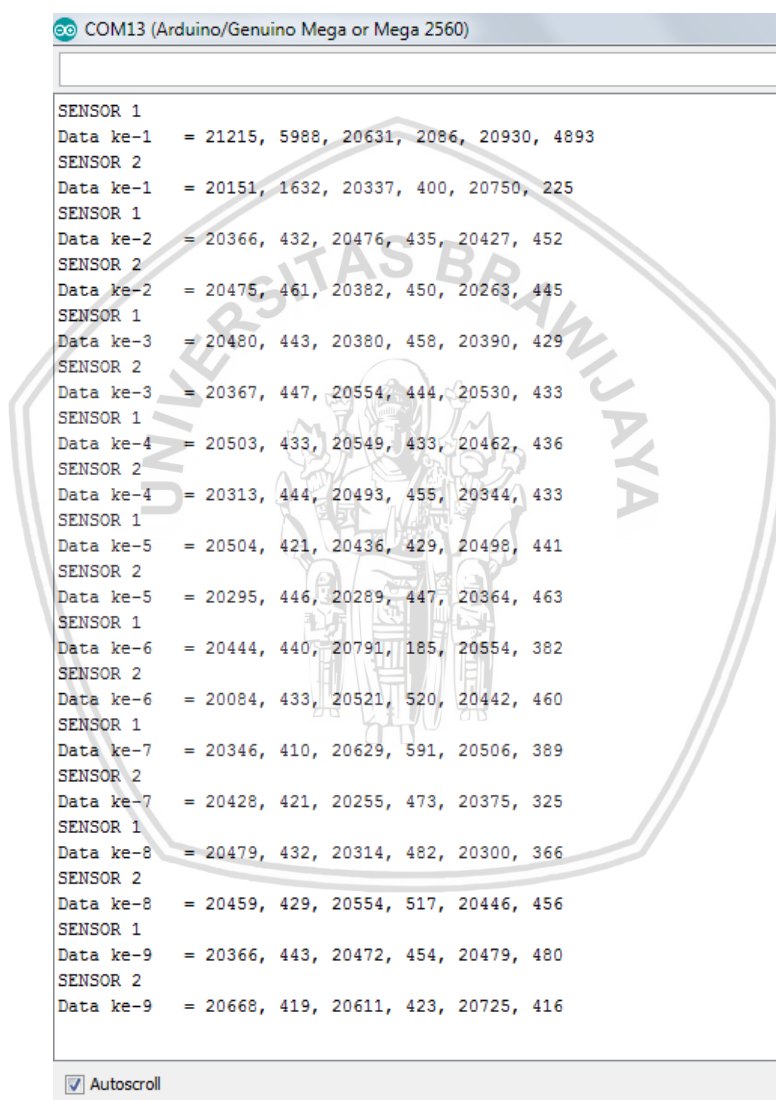
6.1.2.2 Prosedur Pengujian

Berikut prosedur yang dilakukan untuk melakukan pengujian pengiriman dan penerimaan data sensor menggunakan modul komunikasi wireless NRF24L01:

1. Menghubungkan mikrokontroler Arduino Mega yang sudah dirangkai dengan modul komunikasi *wireless* NRF24L01 ke komputer/laptop.

2. Menghubungkan kedua rangkaian pengambilan data sensor dengan baterai 9 volt sebagai daya untuk mikrokontroler Arduino Nano.
3. Membuka aplikasi Arduino IDE dan mengatur *port* Arduino Mega.
4. Meng-*upload* kode program penerimaan nilai sensor ke Arduino Mega. Jika proses upload berhasil, LED pada Arduino Mega akan menyala dan pada Arduino IDE akan muncul keterangan “*done uploading*”.
5. Membuka *Serial Monitor* Arduino IDE.
6. Mengamati hasil pengiriman sensor pada *Serial Monitor* Arduino IDE.

6.1.2.3 Hasil dan Analisis Pengujian



```

COM13 (Arduino/Genuino Mega or Mega 2560)

SENSOR 1
Data ke-1 = 21215, 5988, 20631, 2086, 20930, 4893
SENSOR 2
Data ke-1 = 20151, 1632, 20337, 400, 20750, 225
SENSOR 1
Data ke-2 = 20366, 432, 20476, 435, 20427, 452
SENSOR 2
Data ke-2 = 20475, 461, 20382, 450, 20263, 445
SENSOR 1
Data ke-3 = 20480, 443, 20380, 458, 20390, 429
SENSOR 2
Data ke-3 = 20367, 447, 20554, 444, 20530, 433
SENSOR 1
Data ke-4 = 20503, 433, 20549, 433, 20462, 436
SENSOR 2
Data ke-4 = 20313, 444, 20493, 455, 20344, 433
SENSOR 1
Data ke-5 = 20504, 421, 20436, 429, 20498, 441
SENSOR 2
Data ke-5 = 20295, 446, 20289, 447, 20364, 463
SENSOR 1
Data ke-6 = 20444, 440, 20791, 185, 20554, 382
SENSOR 2
Data ke-6 = 20084, 433, 20521, 520, 20442, 460
SENSOR 1
Data ke-7 = 20346, 410, 20629, 591, 20506, 389
SENSOR 2
Data ke-7 = 20428, 421, 20255, 473, 20375, 325
SENSOR 1
Data ke-8 = 20479, 432, 20314, 482, 20300, 366
SENSOR 2
Data ke-8 = 20459, 429, 20554, 517, 20446, 456
SENSOR 1
Data ke-9 = 20366, 443, 20472, 454, 20479, 480
SENSOR 2
Data ke-9 = 20668, 419, 20611, 423, 20725, 416

☒ Autoscroll
  
```

Gambar 6.2 Hasil Pengujian Pengiriman Data Sensor

Pada Gambar 6.2 dapat dilihat hasil dari pengiriman dan penerimaan data sensor. Data dari Sensor 1 dan Sensor 2 dapat terkirim ke Arduino Mega. Pada hasil pengujian terdapat keterangan “Data ke-” yang menunjukkan jumlah data yang dikirimkan. Dari hasil tersebut dapat diketahui bahwa semua data sensor yang dikirim oleh Arduino Nano dapat diterima oleh Arduino Mega tanpa ada data yang hilang.

6.1.3 Pengujian Tampilan pada *Serial Monitor* Arduino IDE

Serial Monitor merupakan fitur yang disediakan Arduino IDE untuk menampilkan *output* program dan melakukan *input* pada program. *Serial Monitor* dapat diakses menggunakan komunikasi serial, dimana Arduino harus terhubung dengan komputer/laptop menggunakan kabel USB. Pada sistem ini, *output* sistem yang berupa hasil pembacaan sensor dan hasil klasifikasi ditampilkan pada *Serial Monitor* Arduino IDE.

6.1.3.1 Tujuan Pengujian

Pengujian tampilan pada *Serial Monitor* Arduino IDE adalah untuk mengetahui apakah *output* yang ditampilkan pada *Serial Monitor* sudah sesuai dengan perancangan dan implementasi sistem yang diinginkan. Pengujian ini dilakukan dengan mengamati hasil *output* pada *Serial Monitor* sudah sesuai dengan yang ditulis di program atau belum.

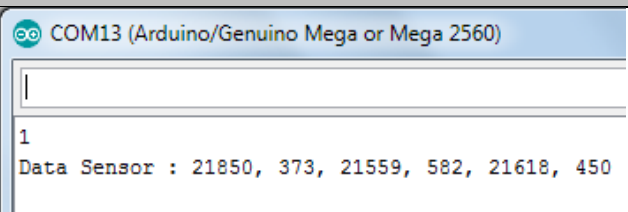
6.1.3.2 Prosedur Pengujian

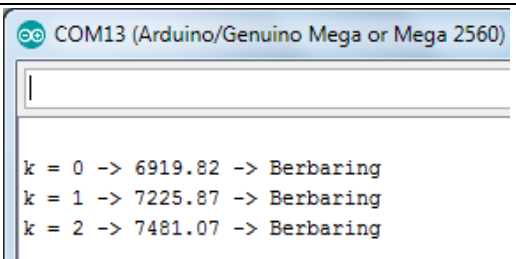
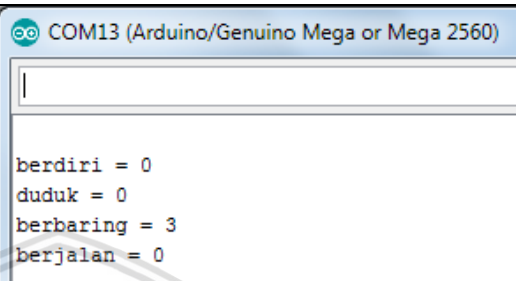
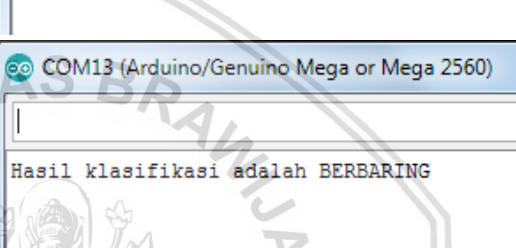
Berikut prosedur yang dilakukan untuk melakukan pengujian tampilan *output* pada *Serial Monitor* Arduino IDE:

1. Menghubungkan mikrokontroler Arduino Mega yang sudah dirangkai dengan modul komunikasi *wireless* NRF24L01 ke komputer/laptop.
2. Menghubungkan rangkaian pengambilan data sensor dengan baterai 9 volt sebagai daya untuk mikrokontroler Arduino Nano.
3. Membuka aplikasi Arduino IDE dan mengatur *port* Arduino Mega.
4. Meng-*upload* kode program penerimaan data dan klasifikasi ke Arduino Mega. Jika proses upload berhasil, LED pada Arduino Mega akan menyala dan pada Arduino IDE akan muncul keterangan "*done uploading*".
5. Membuka *Serial Monitor* Arduino IDE.
6. Mengamati hasil *output* sistem pada *Serial Monitor* Arduino IDE.

6.1.3.3 Hasil dan Analisis Pengujian

Tabel 6.1 Hasil Pengujian Tampilan pada *Serial Monitor* Arduino IDE

No.	Keterangan	Gambar
1	Menampilkan data sensor yang telah diterima.	

2	Menampilkan 3 hasil teratas perhitungan <i>euclidean distance</i> (nilai terkecil) beserta kelasnya.	 <pre> COM13 (Arduino/Genuino Mega or Mega 2560) k = 0 -> 6919.82 -> Berbaring k = 1 -> 7225.87 -> Berbaring k = 2 -> 7481.07 -> Berbaring </pre>
3	Menampilkan hasil perhitungan kelas yang paling banyak muncul terhadap 3 data teratas (proses <i>voting</i> kelas klasifikasi).	 <pre> COM13 (Arduino/Genuino Mega or Mega 2560) berdiri = 0 duduk = 0 berbaring = 3 berjalan = 0 </pre>
4	Menampilkan hasil klasifikasi yang telah dilakukan sistem.	 <pre> COM13 (Arduino/Genuino Mega or Mega 2560) Hasil klasifikasi adalah BERBARING </pre>

Berdasarkan Gambar 6.1 dapat dilihat bahwa tampilan pada Serial Monitor Arduino IDE sudah sesuai dengan perancangan dan implementasi yang telah dilakukan. Sehingga berdasarkan hasil pengujian, dapat dikatakan bahwa pengujian tampilan pada *serial monitor* Arduino IDE berhasil.

6.2 Pengujian Nilai K

Penelitian ini menggunakan metode k-Nearest Neighbor untuk melakukan klasifikasi aktivitas manusia. Metode k-Nearest Neighbor membutuhkan nilai k yang digunakan sebagai konstanta untuk menentukan jumlah tertangga terdekat yang akan diambil dan dicari kelas yang paling banyak muncul.

6.2.1 Tujuan Pengujian

Tujuan pengujian nilai k adalah untuk mendapatkan nilai k yang menghasilkan akurasi tertinggi pada sistem klasifikasi aktivitas manusia. Nilai k dengan akurasi tertinggi akan dijadikan nilai k tetap pada program. Pada pengujian ini, akan dilakukan pengujian dari nilai k sama dengan 1 hingga nilai k sama dengan 10.

6.2.2 Prosedur Pengujian

Berikut prosedur yang dilakukan untuk melakukan pengujian untuk mendapatkan nilai k dengan akurasi tertinggi:

1. Mengambil data aktivitas manusia sebanyak 5 data setiap kelas klasifikasi. Data diambil dari semua posisi peletakan sensor, baik pada

peletakan yang menggunakan satu sensor maupun peletakan yang menggunakan dua sensor. Total posisi peletakan sensor ada 15 posisi, sehingga total data uji yang digunakan untuk pengujian nilai k sebanyak 300 data, karena setiap posisi peletakan sensor terdapat 20 data.

2. Memasukkan data yang digunakan untuk pengujian nilai k pada program sebagai data uji.
3. Mengganti nilai k pada program dari k sama dengan 1 hingga k sama dengan 10.
4. Membuka aplikasi Arduino IDE dan mengatur *port* Arduino Mega.
5. Meng-*upload* kode program klasifikasi pada Arduino Mega. Jika proses upload berhasil, LED pada Arduino Mega akan menyala dan pada Arduino IDE akan muncul keterangan “done uploading”.
6. Membuka *Serial Monitor* Arduino IDE.
7. Mengamati dan mencatat hasil *output* sistem pada *Serial Monitor* Arduino IDE pada Microsoft Excel.
8. Menghitung nilai akurasi hasil klasifikasi dari nilai k sama dengan 1 hingga k sama dengan 10 dengan persamaan berikut:

$$Akurasi = \frac{Total_data - Data_tidak_sesuai}{Total_data} \times 100\% \quad (5.5)$$

9. Mencari nilai k dengan akurasi tertinggi.

6.2.3 Hasil dan Analisis Pengujian

Setelah dilakukan pengujian untuk menentukan nilai k dengan akurasi tertinggi, dilakukan perhitungan akurasi dengan persamaan 5.5. Hasil pengujian nilai k terdapat pada LAMPIRAN. Sementara perbandingan hasil akurasi nilai k dari k sama dengan 1 hingga k sama dengan 10 terdapat pada Gambar 6.3.



Gambar 6.3 Akurasi Hasil Pengujian Nilai k

Pada Gambar 6.3 diketahui bahwa hasil akurasi tertinggi dari pengujian terhadap 300 data dari seluruh posisi peletakan sensor didapatkan bahwa nilai k sama dengan 3 menghasilkan akurasi tertinggi, yaitu sebesar 72,33%. Sehingga

pada implementasi program yang akan digunakan untuk sistem klasifikasi aktivitas manusia, digunakan nilai k sama dengan 3.

6.3 Pengujian Akurasi Sistem Klasifikasi Aktivitas Manusia

Fungsi utama dari sistem ini adalah melakukan klasifikasi aktivitas manusia menggunakan metode *k-Nearest Neighbor*. Pengujian ini dilakukan dengan melakukan aktivitas yang telah ditentukan, yaitu berdiri, duduk, berbaring dan berjalan. Penelitian ini memerlukan 4 orang yang berbeda untuk pengujian dengan satu sensor dan memerlukan 2 orang yang berbeda untuk pengujian dengan dua sensor.

6.3.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui akurasi sistem klasifikasi aktivitas manusia menggunakan metode *k-Nearest Neighbor*. Penelitian ini juga bertujuan untuk mendapatkan posisi peletakan sensor yang paling akurat dari sistem yang menggunakan satu sensor atau sistem yang menggunakan dua sensor.

6.3.2 Prosedur Pengujian

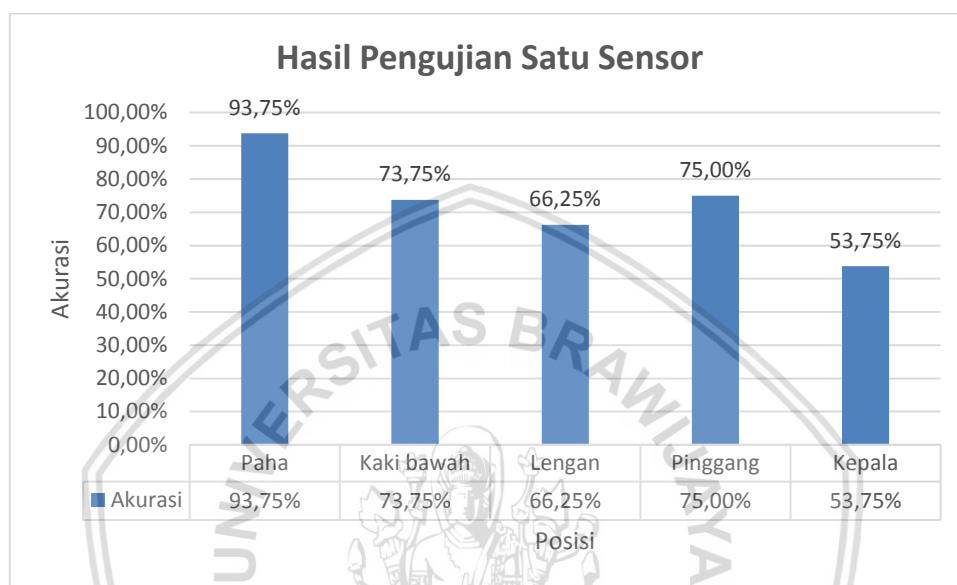
Berikut prosedur yang dilakukan untuk melakukan pengujian akurasi sistem klasifikasi manusia menggunakan metode *k-Nearest Neighbor*:

1. Menghubungkan mikrokontroler Arduino Mega yang sudah dirangkai dengan modul komunikasi *wireless* NRF24L01 ke komputer/laptop.
2. Menghubungkan rangkaian pengambilan data sensor dengan baterai 9 volt sebagai daya untuk mikrokontroler Arduino Nano.
3. Meletakkan rangkaian sensor pada bagian tubuh yang telah dilakukan.
4. Membuka aplikasi Arduino IDE dan mengatur *port* Arduino Mega.
5. Meng-*upload* kode program klasifikasi ke Arduino Mega. Jika proses *upload* berhasil, LED pada Arduino Mega akan menyala dan pada Arduino IDE akan muncul keterangan "*done uploading*".
6. Membuka *Serial Monitor* Arduino IDE.
7. Orang yang menggunakan sensor diminta untuk melakukan aktivitas berdiri, duduk, berbaring dan berjalan secara bergantian. Pengambilan data setiap aktivitas dilakukan sebanyak 5 kali.
8. Setiap orang melakukan pengujian dengan semua posisi peletakan sensor yang berjumlah 5 posisi untuk pengujian dengan satu sensor dan 10 posisi untuk pengujian dengan dua sensor.
9. Mengamati dan mencatat hasil *output* sistem pada *Serial Monitor* Arduino IDE pada Microsoft Excel.
10. Menghitung nilai akurasi hasil klasifikasi dari setiap posisi peletakan sensor dengan persamaan berikut:

$$Akurasi = \frac{Total_data - Data_tidak_sesuai}{Total_data} \times 100\% \quad (5.5)$$

6.3.3 Hasil dan Analisis Pengujian

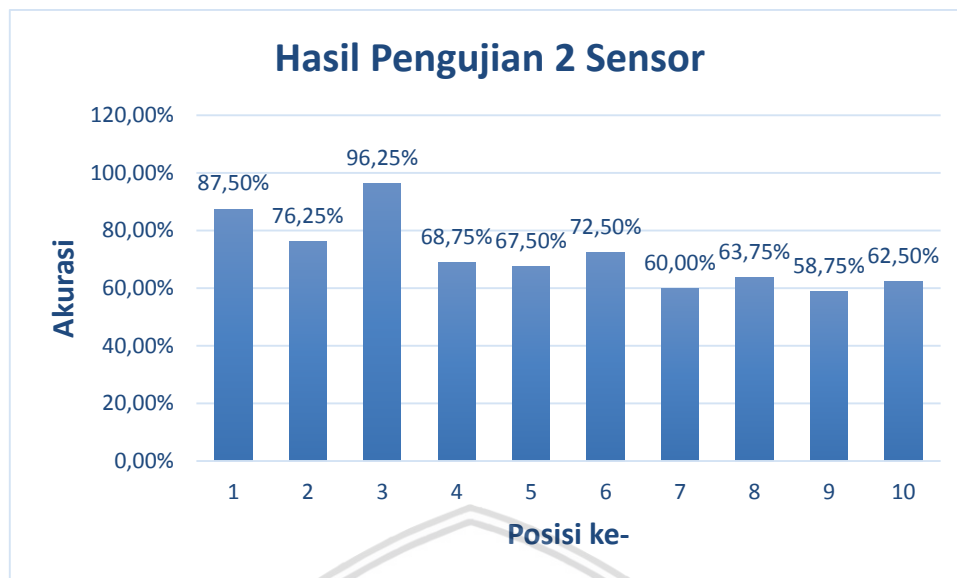
Setelah dilakukan pengujian didapatkan hasil akurasi sistem klasifikasi aktivitas manusia untuk setiap posisi peletakan sensor. Data hasil pengujian secara terperinci terdapat pada LAMPIRAN. Dari hasil pengujian terhadap satu sensor dan dua sensor, didapatkan kesimpulan hasil akurasi yang cukup baik. Gambar 6.4 memperlihatkan total akurasi dari seluruh posisi peletakan sensor menggunakan satu sensor.



Gambar 6.4 Hasil Akurasi Sistem dengan Satu Sensor

Dari Gambar 6.4 dapat disimpulkan bahwa hasil akurasi yang paling tinggi dari satu sensor saat diletakkan di paha dengan akurasi sebesar 93,75%. Sedangkan akurasi terendah adalah posisi peletakan di kepala, yang mendapatkan akurasi sebesar 53,75%. Posisi di paha mendapatkan akurasi tertinggi karena ketika sensor diletakkan di paha, posisi sensor mengalami perubahan dari posisi tegak ketika berdiri menjadi posisi tidur ketika duduk atau berbaring.

Hasil akurasi peletakan sensor yang menggunakan 2 buah sensor terdapat pada Gambar 6.5.



Gambar 6.5 Hasil Akurasi Sistem dengan 2 Sensor

Dari hasil pengujian pada Gambar 6.5, didapatkan hasil bahwa posisi yang paling akurat terletak pada posisi nomor 3 atau di posisi paha dan pinggang yang mendapatkan akurasi sebesar 96,25%. Sedangkan akurasi terendah sebesar 58,75% pada posisi lengan dan kepala. Pada setiap posisi peletakan sensor, terdapat aktivitas yang nilai akurasinya sangat rendah dan ada juga yang nilai akurasinya sangat tinggi. Hal tersebut karena nilai keluaran sensor hampir sama pada 2 aktivitas yang berbeda, sehingga sistem sulit untuk membedakan 2 aktivitas yang nilai sensornya hampir sama.

6.4 Pengujian Waktu Komputasi Klasifikasi k-Nearest Neighbor

Dalam setiap proses klasifikasi, membutuhkan waktu untuk melakukan proses. Pada penelitian ini dilakukan pengujian waktu komputasi klasifikasi k-Nearest Neighbor.

6.4.1 Tujuan Pengujian

Tujuan dilakukan pengujian waktu komputasi klasifikasi k-Nearest Neighbor adalah untuk mengetahui waktu yang dibutuhkan sistem untuk melakukan klasifikasi aktivitas manusia menggunakan metode k-Nearest Neighbor. Waktu komputasi dari sebuah sistem sangat penting karena memengaruhi performansi sistem.

6.4.2 Prosedur Pengujian

Pengujian waktu komputasi sistem dilakukan dengan cara menghitung waktu dari sistem menerima semua data sensor hingga selesai melakukan klasifikasi, yaitu ketika sistem menampilkan hasil klasifikasi. Pengujian waktu komputasi dilakukan pada sistem yang menggunakan satu sensor dan pada sistem yang menggunakan dua sensor. Masing-masing pengujian dilakukan sebanyak 20 kali. Kemudian seluruh hasil pengujian dirata-rata untuk mendapatkan rata-rata waktu

komputasi klasifikasi *k-Nearest Neighbor*. Untuk mendapatkan waktu komputasi sistem, digunakan fungsi `millis()` pada kode program klasifikasi aktivitas manusia. Fungsi `millis()` menghasilkan waktu dalam format *milisecond*. Berikut kode program yang digunakan untuk melakukan pengujian waktu komputasi metode *k-Nearest Neighbor* menggunakan fungsi `millis()`.

Tabel 6.2 Kode Program Fungsi `millis()`

No	Kode Program
1	<code>unsigned long time1, time2, waktu;</code>
2	<code>void loop() {</code>
3	<code> time1 = millis();</code>
4	
5	<code> </code>
6	
7	<code> //Klasifikasi k-Nearest Neighbor</code>
8	
9	<code> </code>
10	
11	
12	<code> time2 = millis();</code>
13	<code> waktu = time2 - time1;</code>
14	<code> Serial.print("Waktu komputasi : ");</code>
15	<code> Serial.print(waktu);</code>
16	<code> Serial.println(" ms");</code>

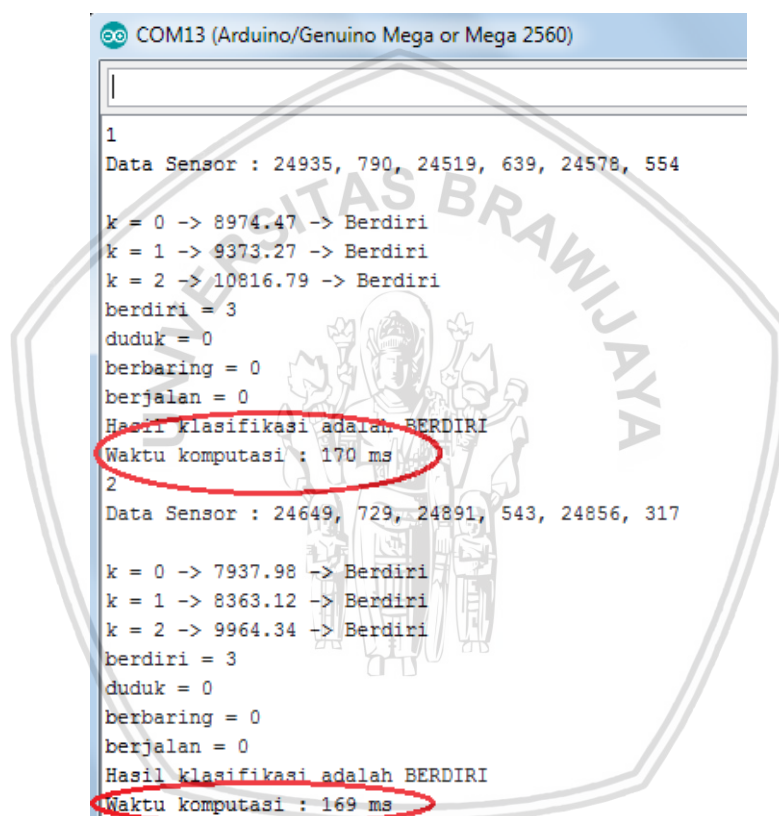
6.4.3 Hasil dan Analisis Pengujian

Tabel 6.3 Hasil Pengujian Waktu Komputasi

No. Pengujian	Waktu yang dibutuhkan	
	Klasifikasi satu sensor	Klasifikasi 2 sensor
Pengujian ke-1	170 ms	353 ms
Pengujian ke-2	169 ms	353 ms
Pengujian ke-3	174 ms	353 ms
Pengujian ke-4	172 ms	354 ms
Pengujian ke-5	175 ms	353 ms
Pengujian ke-6	170 ms	354 ms
Pengujian ke-7	173 ms	353 ms
Pengujian ke-8	175 ms	353 ms
Pengujian ke-9	170 ms	353 ms
Pengujian ke-10	174 ms	353 ms
Pengujian ke-11	174 ms	353 ms
Pengujian ke-12	176 ms	353 ms
Pengujian ke-13	176 ms	354 ms
Pengujian ke-14	174 ms	353 ms
Pengujian ke-15	176 ms	353 ms
Pengujian ke-16	174 ms	353 ms
Pengujian ke-17	174 ms	353 ms
Pengujian ke-18	176 ms	353 ms

Pengujian ke-19	176 ms	353 ms
Pengujian ke-20	174 ms	354 ms
Rata-rata	173,6 ms	353,2 ms

Berdasarkan hasil pengujian pada Tabel 6.18, didapatkan hasil rata-rata waktu komputasi metode k-Nearest Neighbor untuk klasifikasi aktivitas manusia. Pengujian yang dilakukan sebanyak 20 kali menghasilkan rata-rata waktu komputasi metode k-Nearest Neighbor sebesar 173,6 ms atau 0,174 detik untuk pengujian yang menggunakan satu sensor. Hasil tampilan salah satu pengujian waktu komputasi metode k-Nearest Neighbor yang menggunakan satu sensor terdapat pada Gambar 6.6.



```

COM13 (Arduino/Genuino Mega or Mega 2560)

1
Data Sensor : 24935, 790, 24519, 639, 24578, 554

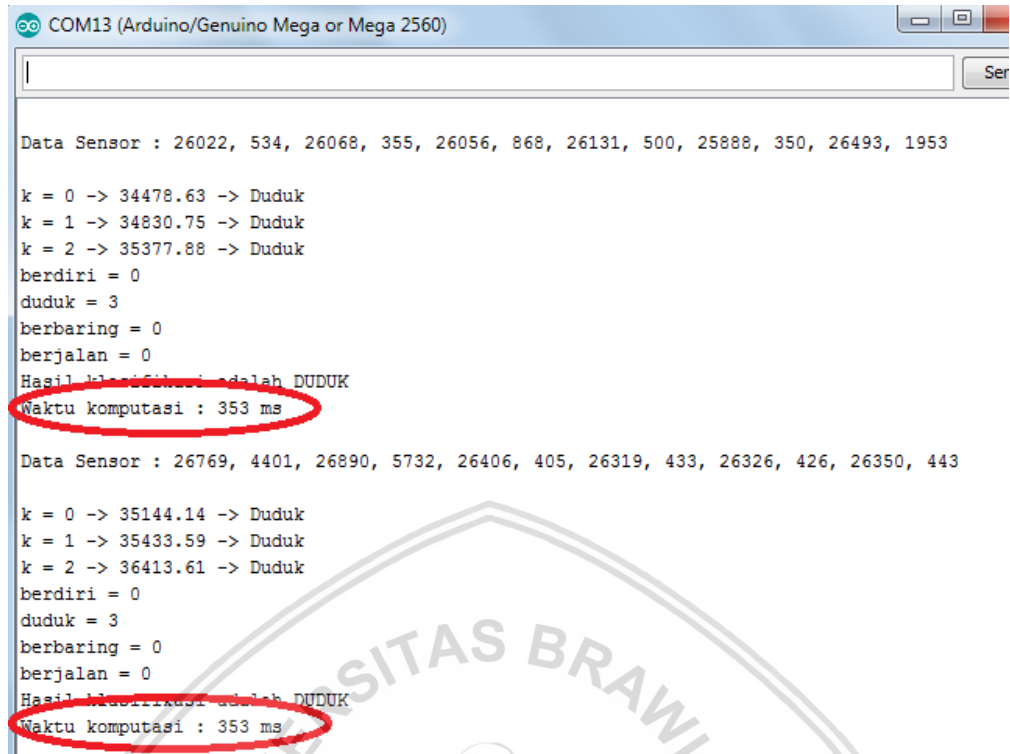
k = 0 -> 8974.47 -> Berdiri
k = 1 -> 9373.27 -> Berdiri
k = 2 -> 10816.79 -> Berdiri
berdiri = 3
duduk = 0
berbaring = 0
berjalan = 0
Hasil klasifikasi adalah BERDIRI
Waktu komputasi : 170 ms

2
Data Sensor : 24649, 729, 24891, 543, 24856, 317

k = 0 -> 7937.98 -> Berdiri
k = 1 -> 8363.12 -> Berdiri
k = 2 -> 9964.34 -> Berdiri
berdiri = 3
duduk = 0
berbaring = 0
berjalan = 0
Hasil klasifikasi adalah BERDIRI
Waktu komputasi : 169 ms
  
```

Gambar 6.6 Hasil Pengujian Waktu Komputasi dengan Satu Sensor

Pada Tabel 6.18 dapat dilihat bahwa pada pengujian waktu komputasi metode k-Nearest Neighbor untuk klasifikasi aktivitas manusia menggunakan 2 sensor didapatkan hasil rata-rata waktu komputasi sebesar 353,2 ms atau 0,353 detik. Pengujian juga dilakukan sebanyak 20 kali untuk menghitung rata-rata waktu komputasi. Hasil tampilan salah satu pengujian waktu komputasi metode k-Nearest Neighbor dengan 2 sensor terdapat pada Gambar 6.7.



```

COM13 (Arduino/Genuino Mega or Mega 2560)

Data Sensor : 26022, 534, 26068, 355, 26056, 868, 26131, 500, 25888, 350, 26493, 1953

k = 0 -> 34478.63 -> Duduk
k = 1 -> 34830.75 -> Duduk
k = 2 -> 35377.88 -> Duduk
berdiri = 0
duduk = 3
berbaring = 0
berjalan = 0
Hasil klasifikasi adalah DUDUK
Waktu komputasi : 353 ms

Data Sensor : 26769, 4401, 26890, 5732, 26406, 405, 26319, 433, 26326, 426, 26350, 443

k = 0 -> 35144.14 -> Duduk
k = 1 -> 35433.59 -> Duduk
k = 2 -> 36413.61 -> Duduk
berdiri = 0
duduk = 3
berbaring = 0
berjalan = 0
Hasil klasifikasi adalah DUDUK
Waktu komputasi : 353 ms
  
```

Gambar 6.7 Hasil Pengujian Waktu Komputasi dengan 2 Sensor

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dibuat dari rumusan masalah, hasil perancangan, implementasi perangkat keras serta perangkat lunak dan pengujian analisis yang dilakukan, maka dapat disimpulkan sebagai berikut:

1. Pada penelitian ini telah dibuat rancangan sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode *k-Nearest Neighbor* berbasis Arduino. Proses perancangan dimulai dengan merancang komponen-komponen perangkat keras untuk membuat rangkaian akuisisi data dan rangkaian penerima data dengan membuat rancangan koneksi antar pin komponen. Rangkaian akuisisi data digunakan untuk mengambil nilai sensor *accelerometer* dan *gyroscope* kemudian mengirimkannya menggunakan modul komunikasi *wireless* NRF24L01. Sensor yang digunakan adalah sensor MPU6050 yang sudah terdapat sensor *accelerometer* dan *gyroscope* di dalamnya. Rangkaian penerima data digunakan untuk menerima data sensor yang telah dikirimkan oleh rangkaian akuisisi data serta melakukan klasifikasi terhadap data tersebut. Setelah itu dilakukan perancangan perangkat lunak berupa perangkat lunak untuk sistem akuisisi data, perangkat lunak untuk sistem penerima data dan perangkat lunak metode *k-Nearest Neighbor*.
2. Implementasi sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode *k-Nearest Neighbor* berbasis Arduino dilakukan dengan mengimplementasikan komponen-komponen perangkat keras yang dirangkai pada PCB sesuai dengan perancangan sistem perangkat keras. Program-program yang dibutuhkan diimplementasikan sesuai dengan algoritma atau diagram alir yang telah dirancang. Kemudian rangkaian akuisisi data yang berisi sensor diletakan pada bagian tubuh yang telah ditentukan, yaitu kepala, lengan, pinggang, paha dan kaki bagian bawah.
3. Hasil pengujian fungsional sistem klasifikasi aktivitas manusia menggunakan sensor *accelerometer* dan *gyroscope* dengan metode *k-Nearest Neighbor* berbasis Arduino menunjukkan bahwa sistem sudah berjalan sesuai dengan perancangan dan implementasi. Didapatkan pula hasil akurasi tertinggi sebesar 93,75% untuk sistem yang menggunakan satu sensor dengan posisi peletakan sensor berada di paha. Sedangkan akurasi terendah untuk sistem yang menggunakan satu sensor sebesar 53,75% dengan posisi peletakan sensor berada di kepala. Pada sistem yang menggunakan dua sensor didapatkan hasil akurasi tertinggi sebesar 96,25% dengan peletakan sensor berada di paha dan pinggang. Sedangkan akurasi terendah sebesar 58,75% dengan peletakan sensor berada di lengan dan kepala.
4. Hasil pengujian waktu komputasi metode *k-Nearest Neighbor* dalam melakukan klasifikasi aktivitas manusia didapatkan rata-rata waktu komputasi sebesar 173,6 ms atau 0,174 detik untuk pengujian yang menggunakan satu sensor dari 20 kali pengujian. Untuk pengujian yang menggunakan dua sensor

didapatkan rata-rata waktu komputasi metode *k-Nearest Neighbor* dalam melakukan klasifikasi aktivitas manusia sebesar 353,2 ms atau 0,353 detik dari 20 kali pengujian.

7.2 Saran

Saran agar sistem ini bisa dikembangkan untuk penelitian selanjutnya yaitu:

1. Diharapkan sistem ini dapat diimplementasikan untuk tujuan tertentu, misalkan untuk *smart home* atau *monitoring* kesehatan.
2. Menambah jumlah data latih sebanyak mungkin sehingga mengurangi tingkat persentase kesalahan pada sistem.
3. Disarankan untuk menggunakan metode pengklasifikasian lain, untuk dapat membandingkan hasil akurasi dari tiap metode.



DAFTAR PUSTAKA

- Arduino, 2016. *Arduino Mega ADK*. [Online]
Available at: <https://www.arduino.cc/en/Main/ArduinoBoardMegaADK?from=Main.ArduinoBoardADK>
[Diakses 14 Mei 2018].
- Arduino, 2016. *Arduino Nano*. [Online]
Available at: <https://www.arduino.cc/en/Main/ArduinoBoardNano?setlang=it>
[Diakses 13 Mei 2018].
- Arduino, 2016. *MPU-6050 Accelerometer + Gyro*. [Online]
Available at: <http://playground.arduino.cc/Main/MPU-6050>
[Diakses 13 Mei 2018].
- Berlia, R. & Santosh, P., 2014. Mouse Brace (A convenient computer mouse using accelerometer, flex sensors and microcontroller). *IEEE*, pp. 558-561.
- Cao, L., Wang, Y., Jin, Q. & Ma, J., 2017. ActiRecognizer: Design and implementation of a realtime human activity recognition system. *IEEE*, Volume 71, p. 266.
- Dejan, 2017. *Arduino Wireless Communication – NRF24L01 Tutorial*. [Online]
Available at: <https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/>
[Diakses 13 Mei 2018].
- Elektroku, 2017. *Pengertian, Prinsip Kerja Dan Kegunaan Sensor Gyroscope*. [Online]
Available at: <https://elektroku.com/pengertian-prinsip-kerja-dan-kegunaan-sensor-gyroscope/>
[Diakses 13 Mei 2018].
- Hasani, S. A., 2015. *Klasifikasi Status Gizi Menggunakan K-nearest Neighbor. Klasifikasi Status Gizi Menggunakan K-nearest Neighbor*, Juli.
- Immersa, 2018. *Pengertian Accelerometer dan Cara Kerjanya*. [Online]
Available at: <http://www.immersa-lab.com/pengertian-accelerometer-dan-cara-kerjanya.htm>
[Diakses 10 Mei 2018].
- Immersa, 2018. *Pengertian Gyroscope dan Cara Kerjanya*. [Online]
Available at: <http://www.immersa-lab.com/pengertian-gyroscope-dan-cara-kerjanya.htm>
[Diakses 10 Mei 2018].
- Invensense, 2012. *MPU-6050 Datasheet (PDF) - List of Unclassified Manufacturers*. [Online]
Available at: <http://pdf1.alldatasheet.com/datasheet->

- <pdf/view/517744/ETC1/MPU-6050.html>
[Diakses 15 Mei 2018].
- Ivan, 2017. *Inovasi teknologi yang mengubah masa depan*. [Online]
Available at: <https://beritagar.id/artikel/sains-teknologi-inovasi-teknologi-yang-mengubah-masa-depan>
[Diakses 10 April 2018].
- Jain, A. & Kanhangad, V., 2018. Human Activity Classification in Smartphones Using Accelerometer and Gyroscope Sensors. *IEEE*, 1 February, 18(3), p. 1169.
- Ndaumanu, R. I., 2014. Analisis Prediksi Tingkat Pengunduran Diri Mahasiswa dengan Metode K-Nearest Neighbor. *JatISI*, 1(1), pp. 1-15.
- Norhabibah, S., 2016. Rancang Bangun Sistem Monitoring Deteksi Jatuh untuk Manula dengan Menggunakan Accelerometer. *Journal of Informatics, Network, and Computer Science*, pp. 43-52.
- Oktaviani, G., 2016. Algoritma K-Nearest Neighbor Classification Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa. *Jurnal Sains, Teknologi dan Industri*, 13(2), pp. 195-202.
- Putra, S., 2016. Identifikasi Aktivitas Manusia di Dalam Ruangan Menggunakan IP Camera dengan Metode Template Matching dan Algoritma Klasifikasi. *ResearchGate*.
- SM Team, 2017. *Cari Tahu Apa Bedanya Supervised vs Unsupervised Learning*. [Online]
Available at: <https://www.jagoanhosting.com/blog/cari-tahu-apa-bedanya-supervised-vs-unsupervised-learning/>
[Diakses 2 Mei 2018].
- Srikanth, G., Sudarshan, K. & Gopinath, R., 2012. *Human Activity Recognition*. [Online]
Available at: <https://www.slideshare.net/srikanthgadam/human-activity-recognition>
[Diakses 2 Mei 2018].
- Sukma, A. et al., 2014. *web.unair.ac.id*. [Online]
Available at: http://web.unair.ac.id/admin/file/f_41382_STKI-KEL-2_K-NEAREST-NEIGHBOR.pdf
[Diakses 23 April 2018].
- Su, X., 2014. *Activity Recognition with Smartphone Sensors Second Exam Literature Review*, New York: The City University of New York Graduate Center.
- Zubair, M., 2016. Human Activity Recognition Using Wearable Accelerometer Sensors. *IEEE International Conference on Consumer Electronics-Asia*, Volume 1, p. 978.

LAMPIRAN A DATA LATIH

A.1 DATA LATIH SISTEM DENGAN SATU SENSOR

A.1.1 PELETAKAN SENSOR DI PAHA

No	tot_acc1 (fitur 1)	tot_gy1 (fitur2)	tot_acc2 (fitur 3)	tot_gy2 (fitur4)	tot_acc3 (fitur 5)	tot_gy3 (fitur6)	Kelas Klasifikasi
1	23443	24764	25177	3227	26567	17071	BERDIRI
2	25150	3339	25143	2089	25030	1479	BERDIRI
3	25235	551	25206	1207	25323	454	BERDIRI
4	25847	4145	24835	13252	25572	1664	BERDIRI
5	25212	317	25403	3354	25398	2635	BERDIRI
6	15681	609	15675	452	15697	275	BERDIRI
7	12859	15024	15561	977	15517	1813	BERDIRI
8	15678	453	15753	636	15492	594	BERDIRI
9	15438	1973	15341	685	15490	496	BERDIRI
10	15490	1144	15535	456	15226	4035	BERDIRI
11	16787	6369	18132	1445	17787	842	DUDUK
12	17235	12343	17542	1424	17667	969	DUDUK
13	17762	468	17728	287	17753	372	DUDUK
14	17766	645	17712	410	18947	4461	DUDUK
15	17573	305	17503	388	17531	438	DUDUK
16	17829	665	17710	314	17848	384	DUDUK
17	17774	397	17637	303	17663	391	DUDUK
18	17664	1335	17808	693	17699	637	DUDUK
19	17187	655	17432	527	17381	651	DUDUK
20	17827	256	17661	891	17557	852	DUDUK
21	22465	724	21106	1200	22267	4720	BERBARING
22	17460	789	17652	255	17882	690	BERBARING
23	22248	1722	22251	561	22162	420	BERBARING
24	17460	789	17652	255	17882	690	BERBARING
25	22154	526	22101	436	22224	402	BERBARING
26	22070	342	22089	407	22131	241	BERBARING
27	17874	264	17500	354	17598	2364	BERBARING
28	22187	494	22093	381	22113	425	BERBARING
29	22131	455	22231	295	22247	445	BERBARING
30	17725	245	17745	526	17638	284	BERBARING
31	25672	31681	25038	1688	24961	2706	BERJALAN
32	25054	526	25496	7776	24452	14410	BERJALAN
33	19536	10805	16857	14559	17523	16167	BERJALAN
34	26302	23102	25130	13305	25915	2491	BERJALAN
35	25158	9746	24448	6461	24913	5658	BERJALAN
36	25836	11695	25676	14140	25890	4174	BERJALAN
37	18480	10413	16185	8232	16002	2761	BERJALAN
38	25333	3073	25607	15145	26307	15946	BERJALAN
39	32584	26715	24515	5743	24581	-32647	BERJALAN
40	24258	6711	26034	21406	25832	17071	BERJALAN

A.1.2 PELETAKAN SENSOR DI KAKI BAGIAN BAWAH

No	tot_acc1 (fitur 1)	tot_gy1 (fitur2)	tot_acc2 (fitur 3)	tot_gy2 (fitur4)	tot_acc3 (fitur 5)	tot_gy3 (fitur6)	Kelas Klasifikasi
1	16107	1405	15995	801	16237	624	BERDIRI
2	16142	189	16054	296	16023	915	BERDIRI
3	16176	159	16054	186	16122	520	BERDIRI
4	16203	409	16106	226	16200	488	BERDIRI
5	16193	4011	16431	2570	16535	2708	BERDIRI
6	25636	3508	26551	9191	25130	4209	BERDIRI
7	25734	1871	25721	851	25728	752	BERDIRI
8	25587	597	25544	682	25620	1138	BERDIRI
9	25651	2161	25601	807	25698	567	BERDIRI
10	25784	400	25695	528	25733	352	BERDIRI
11	25865	2264	26046	2303	26314	1870	DUDUK
12	15706	1514	15887	730	16048	720	DUDUK
13	26047	521	26205	410	25982	633	DUDUK
14	25922	1210	26025	2882	26063	476	DUDUK
15	15984	865	16099	1982	15953	430	DUDUK
16	25998	480	26070	1020	25959	511	DUDUK
17	26151	836	26057	549	26145	857	DUDUK
18	26250	2259	26400	2644	26105	4139	DUDUK
19	15620	576	15845	1821	15802	1815	DUDUK
20	26448	3006	26091	2196	25968	422	DUDUK
21	22096	411	21801	871	23052	3547	BERBARING
22	17667	440	17313	789	17668	1244	BERBARING
23	22095	482	22122	600	22037	399	BERBARING
24	17725	386	17551	209	17536	252	BERBARING
25	22079	517	22075	1072	22134	380	BERBARING
26	17568	578	17622	246	17653	221	BERBARING
27	22000	399	21948	469	22085	392	BERBARING
28	22117	416	22154	433	22166	423	BERBARING
29	17677	275	17578	233	17530	243	BERBARING
30	22170	407	22103	360	22275	482	BERBARING
31	18470	15154	17574	-19184	19776	29469	BERJALAN
32	26907	25310	26338	17896	23797	10025	BERJALAN
33	22130	-25058	25277	28855	26012	-32412	BERJALAN
34	15266	3757	15680	32123	15118	22856	BERJALAN
35	25244	25909	22579	27273	22724	28003	BERJALAN
36	19924	-8782	16614	-21069	11573	-27121	BERJALAN
37	24574	4903	24964	19656	22222	-19086	BERJALAN
38	17347	-20341	18000	19183	15788	26868	BERJALAN
39	25580	5508	23996	7690	25128	10411	BERJALAN
40	14794	2852	15905	231	15903	417	BERJALAN

A.1.3. PELETAKAN SENSOR DI LENGAN

No	tot_acc1 (fitur 1)	tot_gy1 (fitur2)	tot_acc2 (fitur 3)	tot_gy2 (fitur4)	tot_acc3 (fitur 5)	tot_gy3 (fitur6)	Kelas Klasifikasi
1	27419	7335	26424	1391	26299	3887	BERDIRI
2	26964	674	26044	837	26581	2693	BERDIRI
3	26468	2789	25463	3236	26319	3606	BERDIRI
4	26270	960	26923	1785	26608	2187	BERDIRI
5	24710	2642	18546	14925	29188	17952	BERDIRI
6	15772	4863	15685	2504	19137	30457	BERDIRI
7	15683	2198	16077	15773	16070	2545	BERDIRI
8	15907	614	15757	2808	15875	1569	BERDIRI
9	15867	2767	15644	404	15845	2942	BERDIRI
10	15783	353	15917	434	15793	728	BERDIRI
11	15611	2143	15886	1133	15868	2462	DUDUK
12	27497	-32660	26833	13429	25464	20510	DUDUK
13	16705	12954	15776	2908	16162	6802	DUDUK
14	26547	445	26334	668	26243	1331	DUDUK
15	26193	481	26174	629	26123	714	DUDUK
16	15897	606	15932	646	15891	267	DUDUK
17	26071	601	26158	574	26428	3386	DUDUK
18	15887	5319	15928	790	15797	2191	DUDUK
19	25548	4000	26045	1186	25437	8294	DUDUK
20	15713	1702	15845	321	15857	1037	DUDUK
21	25615	3714	25481	2012	24643	12402	BERBARING
22	15412	2543	20559	12512	19655	12502	BERBARING
23	16261	13016	15541	1332	15694	13172	BERBARING
24	19443	6600	22958	15316	23829	2100	BERBARING
25	16094	675	15848	259	16076	136	BERBARING
26	24163	424	24141	385	23843	852	BERBARING
27	24561	639	23950	481	23852	412	BERBARING
28	15911	595	15803	237	15868	178	BERBARING
29	23995	470	23863	490	24310	241	BERBARING
30	24006	364	24090	377	23914	541	BERBARING
31	23950	9515	25892	28516	24993	-19118	BERJALAN
32	21157	-27636	15074	19105	16620	-31053	BERJALAN
33	29524	-25051	25994	-25921	28688	14571	BERJALAN
34	14638	16061	16990	20653	14888	25359	BERJALAN
35	22217	20487	32268	-32490	24583	4381	BERJALAN
36	31008	5832	26507	26388	24071	27148	BERJALAN
37	16117	14148	18119	10377	14615	18430	BERJALAN
38	25016	-19628	28092	-29851	26249	-29435	BERJALAN
39	26281	13872	24919	15802	22668	4850	BERJALAN
40	26315	2945	27561	11661	24694	2874	BERJALAN

A.1.4 PELETAKAN SENSOR DI PINGGANG

No	tot_acc1 (fitur 1)	tot_gy1 (fitur2)	tot_acc2 (fitur 3)	tot_gy2 (fitur4)	tot_acc3 (fitur 5)	tot_gy3 (fitur6)	Kelas Klasifikasi
1	23344	1478	23510	2153	23343	5133	BERDIRI
2	23304	10096	23408	662	23764	831	BERDIRI
3	22794	4994	23785	982	23693	462	BERDIRI
4	23713	267	23837	568	23311	1581	BERDIRI
5	27995	5937	25555	3249	23659	895	BERDIRI
6	16068	1059	16357	822	16183	662	BERDIRI
7	16514	1982	16087	1322	15625	2727	BERDIRI
8	16127	485	16090	350	16139	250	BERDIRI
9	15986	376	15977	517	16071	615	BERDIRI
10	15969	512	15912	1154	15908	448	BERDIRI
11	6768	4099	11239	6943	7032	561	DUDUK
12	16097	497	16264	1292	16294	3085	DUDUK
13	7210	3119	7341	569	7643	607	DUDUK
14	16345	958	16386	1927	16240	423	DUDUK
15	7782	696	7402	1008	7181	659	DUDUK
16	7228	397	7294	600	7054	1307	DUDUK
17	7453	514	7393	480	7470	391	DUDUK
18	16479	334	16431	394	16497	248	DUDUK
19	6771	6790	8636	2018	9702	943	DUDUK
20	16345	3326	16072	988	15975	1064	DUDUK
21	16462	491	16397	461	16509	195	BERBARING
22	23186	658	23275	530	23153	517	BERBARING
23	23446	611	23199	549	23186	563	BERBARING
24	26772	15272	25751	1499	25415	609	BERBARING
25	16426	554	16738	361	16486	90	BERBARING
26	25502	741	25727	404	25482	550	BERBARING
27	16477	291	16605	410	16408	353	BERBARING
28	25382	341	25512	802	25478	517	BERBARING
29	23168	516	23292	461	23269	700	BERBARING
30	16423	451	16496	327	16496	145	BERBARING
31	25456	4836	26962	10569	26453	1660	BERJALAN
32	17029	11235	15649	7089	16420	3730	BERJALAN
33	26488	4833	26376	5679	24021	9185	BERJALAN
34	18866	25912	14776	6795	19853	15341	BERJALAN
35	25342	3033	27167	5748	25712	3442	BERJALAN
36	26453	2951	26376	4087	25902	5245	BERJALAN
37	14876	2395	15799	6865	16781	7680	BERJALAN
38	26269	1731	28264	6434	27531	8994	BERJALAN
39	25305	3699	25339	5883	25421	6887	BERJALAN
40	16257	12096	15906	9163	16531	4998	BERJALAN

A.1.5 PELETAKAN SENSOR DI KEPALA

No	tot_acc1 (fitur 1)	tot_gy1 (fitur2)	tot_acc2 (fitur 3)	tot_gy2 (fitur4)	tot_acc3 (fitur 5)	tot_gy3 (fitur6)	Kelas Klasifikasi
1	17465	516	17166	227	17486	338	BERDIRI
2	17322	284	17439	274	17465	355	BERDIRI
3	17615	420	17403	142	17442	437	BERDIRI
4	17368	398	17443	312	17331	499	BERDIRI
5	17479	490	17203	387	17428	202	BERDIRI
6	25680	467	26087	1349	25580	337	BERDIRI
7	25670	510	25740	826	25757	422	BERDIRI
8	25151	1492	24946	1627	25347	734	BERDIRI
9	25254	426	25111	689	25353	560	BERDIRI
10	25312	751	25338	356	25617	562	BERDIRI
11	25106	399	25090	387	25126	317	DUDUK
12	25157	464	25162	335	25097	372	DUDUK
13	24996	165	25133	369	25085	518	DUDUK
14	25049	508	25025	541	25188	523	DUDUK
15	24331	464	24667	478	25568	449	DUDUK
16	17450	4516	18386	7541	18386	3797	DUDUK
17	17744	1812	17443	500	17408	409	DUDUK
18	17671	2528	17635	1493	17524	814	DUDUK
19	17595	458	17568	534	17574	431	DUDUK
20	17783	4332	16939	1732	17447	657	DUDUK
21	19821	490	19997	946	20003	419	BERBARING
22	19796	381	19876	527	20043	495	BERBARING
23	19922	422	19920	344	19851	471	BERBARING
24	20478	7105	18706	15702	22952	10342	BERBARING
25	29537	5062	24473	2840	28250	6920	BERBARING
26	16488	2535	18118	9248	17067	718	BERBARING
27	16256	1334	17197	288	16955	523	BERBARING
28	17503	1497	16847	243	16683	134	BERBARING
29	17188	192	16520	508	16539	138	BERBARING
30	16601	384	16404	299	16859	357	BERBARING
31	22494	2305	23018	2125	22877	3776	BERJALAN
32	21234	4235	22060	1801	25593	7508	BERJALAN
33	22742	8581	21254	11174	24902	970	BERJALAN
34	26172	4193	25716	3286	32410	24861	BERJALAN
35	29387	3344	24509	3543	26181	923	BERJALAN
36	17045	10566	18080	10669	17360	7356	BERJALAN
37	17075	926	15963	2027	15384	2096	BERJALAN
38	17082	2335	16234	7682	16447	1847	BERJALAN
39	17728	3111	17410	3610	17234	4661	BERJALAN
40	18017	8470	16450	10580	16512	6372	BERJALAN

A.2 DATA LATIH SISTEM DENGAN DUA SENSOR

A.2.1 PELETAKAN SENSOR DI PAHA DAN KAKI BAGIAN BAWAH

No	tot_acc1 sensor 1 (fitur 1)	tot_gy1 sensor 1 (fitur2)	tot_acc2 sensor 1 (fitur 3)	tot_gy2 sensor 1 (fitur4)	tot_acc3 sensor 1 (fitur 5)	tot_gy3 sensor 1 (fitur6)	tot_acc1 sensor 2 (fitur 7)	tot_gy1 sensor 2 (fitur 8)	tot_acc2 sensor 2 (fitur 9)	tot_gy2 sensor 2 (fitur10)	tot_acc3 sensor 2 (fitur11)	tot_gy3 sensor 2 (fitur12)	Kelas Klasifikasi
1	24865	1865	24845	7290	24693	2396	15464	4819	15357	4871	15514	2790	BERDIRI
2	24894	897	25028	1696	24449	12727	15999	4354	16796	4420	15949	236	BERDIRI
3	26217	21895	26261	1306	25123	3404	15202	10595	14537	6892	17115	15990	BERDIRI
4	25000	1562	24777	2467	24814	775	15614	2270	15518	400	15548	312	BERDIRI
5	24844	783	24735	333	24828	471	15541	1818	15406	291	15413	868	BERDIRI
6	23729	27821	24195	12673	24480	6558	15331	4457	15750	975	15776	255	BERDIRI
7	23470	19814	24680	3432	25087	2087	16395	2263	18430	2958	15693	8607	BERDIRI
8	24832	6718	25096	10289	24538	2953	15832	18344	15726	21735	15798	4810	BERDIRI
9	24680	743	24732	558	24720	518	15407	9196	15368	1468	15420	414	BERDIRI
10	24706	2664	24899	1662	24821	4368	15140	4725	15171	2590	15065	4839	BERDIRI
11	25036	1687	24581	3297	24750	2246	15108	1529	15889	5644	15026	7969	DUDUK
12	24422	705	24837	357	24576	174	15772	5143	15202	1546	15617	621	DUDUK
13	24953	454	24786	637	24866	545	15244	309	15343	665	15596	659	DUDUK
14	24775	478	24824	530	25019	123	15857	649	15452	729	15511	642	DUDUK
15	25083	192	24859	518	25009	296	15335	423	15321	1091	15465	874	DUDUK
16	24664	403	24906	340	24829	509	15618	603	15565	1184	15294	1079	DUDUK
17	26100	359	26267	369	26164	480	17012	301	17096	444	17240	377	DUDUK
18	26095	447	26155	445	25990	504	17198	389	17071	481	17321	259	359DUDUK
19	26243	11651	24183	25963	24416	11441	16764	739	17958	397	17156	794	DUDUK
20	25151	3048	25255	4441	25003	1404	16954	372	17269	440	17195	431	DUDUK
21	25168	4012	24536	2982	25506	882	17693	1244	17939	1074	17214	764	BERBARING
22	22558	421	22415	500	22554	507	17730	246	17885	263	17722	221	BERBARING
23	23225	2129	23095	494	23056	2449	22080	23885	19160	29224	17870	5920	BERBARING
24	22890	362	22658	319	22948	634	17776	467	18262	831	17582	671	BERBARING
25	22372	520	22687	4284	22395	2648	17483	1381	18373	3754	17855	1338	BERBARING
26	22960	685	22803	801	22709	397	17564	1201	17251	388	17918	353	BERBARING

27	22784	434	22800	432	22733	422	17737	326	17654	249	17405	500	BERBARING
28	22689	544	22511	520	22455	540	17842	258	17761	300	17862	228	BERBARING
29	22513	387	22571	433	22567	414	17764	254	17919	248	17664	221	BERBARING
30	22460	558	22597	307	22522	497	17774	306	17770	246	17907	395	BERBARING
31	27230	3787	24313	21754	23060	6121	14538	9389	17554	18552	14655	7878	BERJALAN
32	26075	14281	24614	10882	28093	30891	15198	19375	16313	4608	25630	-26702	BERJALAN
33	27592	-32420	26753	3999	24891	13283	22423	-31528	14632	20714	12911	30584	BERJALAN
34	25402	-29294	25123	3104	22685	26560	17829	-23406	16868	-18604	17269	-29203	BERJALAN
35	24819	20472	27420	15823	26330	2340	16344	6570	16539	4905	15751	9370	BERJALAN
36	25757	9838	25368	4337	23782	30404	15592	6622	16417	-25496	9954	15316	BERJALAN
37	27783	24733	25884	-30326	22499	5727	10434	-24330	16841	19996	-32170	-14315	BERJALAN
38	21748	-26041	25975	5702	22855	28317	12110	28048	12596	18527	11289	31425	BERJALAN
39	23235	-17855	30362	17846	27646	-26178	15616	28176	13675	20271	17975,	-27760	BERJALAN
40	24492	16580	30490	-17088	23848	-28049	15616	28176	13675	20271	17975	-27760	BERJALAN

A.1.2 PELETAKAN SENSOR DI PAHA DAN LENGAN

No	tot_acc1 sensor 1 (fitur 1)	tot_gy1 sensor 1 (fitur2)	tot_acc2 sensor 1 (fitur 3)	tot_gy2 sensor 1 (fitur4)	tot_acc3 sensor 1 (fitur 5)	tot_gy3 sensor 1 (fitur6)	tot_acc1 sensor 2 (fitur 7)	tot_gy1 sensor 2 (fitur 8)	tot_acc2 sensor 2 (fitur 9)	tot_gy2 sensor 2 (fitur10)	tot_acc3 sensor 2 (fitur11)	tot_gy3 sensor 2 (fitur12)	Kelas Klasifikasi
1	24965	11655	25315	7000	25637	9536	15983	4171	15970	9285	15930	8867	BERDIRI
2	25602	6784	26190	3939	26166	2468	15626	26434	16375	19170	15559	3533	BERDIRI
3	25695	429	25842	427	25732	478	16000	518	15908	475	15340	3966	BERDIRI
4	26000	1578	25902	334	25674	557	15987	381	15829	333	15857	1137	BERDIRI
5	25625	891	25894	1046	25841	344	15830	373	15793	590	16057	548	BERDIRI
6	25771	481	25777	332	25775	2226	15864	931	15687	1408	15999	388	BERDIRI
7	25724	723	25851	642	25991	600	15916	1503	15477	2962	16594	1634	BERDIRI
8	25792	414	25769	371	25805	681	15925	1256	15861	800	15852	703	BERDIRI
9	25712	631	25831	504	25565	1170	16048	823	15916	453	15970	782	BERDIRI
10	25820	895	25789	2058	25886	569	16365	6184	15751	629	15826	956	BERDIRI
11	23909	481	23859	699	23950	622	15767	290	15627	563	15751	422	DUDUK
12	23851	337	23840	215	23921	498	15869	838	15914	512	15879	596	DUDUK
13	23815	1968	23690	3067	24189	2718	15106	2342	15147	1135	16290	2361	DUDUK
14	24148	1907	24050	1929	24371	3150	16196	5798	15427	911	15734	3431	DUDUK
15	24143	764	24125	372	24105	415	15664	759	15434	983	15476	1052	DUDUK
16	24463	814	24306	1440	24322	2833	15874	1760	14863	531	15492	1483	DUDUK
17	24278	558	24354	476	24386	682	15414	2046	15436	665	15638	887	DUDUK
18	24535	948	24415	713	24493	306	15505	301	15438	677	15500	1149	DUDUK
19	24482	418	24351	354	24326	663	15570	834	15401	476	15443	241	DUDUK
20	24335	651	24384	284	24365	521	15472	343	15378	1888	15564	866	DUDUK
21	24619	3941	24558	6394	26683	13104	17780	3860	16506	5847	17607	833	BERBARING
22	25751	592	25147	13393	24142	4170	17959	1441	17510	821	17643	336	BERBARING
23	26099	4207	26687	445	26548	635	17630	1833	17629	796	17754	213	BERBARING
24	26578	554	26610	537	26512	391	17671	315	17553	204	17579	207	BERBARING
25	26661	720	26616	296	26479	486	17645	282	17755	430	17566	290	BERBARING
26	26286	378	26285	482	26212	464	17589	234	17430	384	17449	230	BERBARING
27	26204	413	26247	435	26213	511	17593	219	17464	348	17466	363	BERBARING
28	26159	414	26139	480	26208	518	17399	278	17498	359	17556	337	BERBARING
29	26345	379	26201	341	26236	501	17783	163	17693	420	17534	270	BERBARING

30	26184	468	26167	350	26223	439	17392	196	17484	405	17588	270	BERBARING
31	27095	-32499	25365	26472	23277	16908	20464	-19137	22808	26566	17441	13096	BERJALAN
32	24424	5796	22636	8537	27250	13670	15503	8959	23995	24203	12078	23689	BERJALAN
33	24246	8090	28720	3939	21770	28807	13736	11165	20089	6026	13703	5082	BERJALAN
34	25303	15597	26324	16878	22490	8244	17706	17525	15479	16369	22813	26092	BERJALAN
35	27833	1778	25283	-30425	19556	11992	15225	-16725	16677	-24014	12637	28753	BERJALAN
36	23611	24956	23772	11651	21540	3548	13468	30018	18412	-26773	18735	11167	BERJALAN
37	24588	-23631	23997	-26484	25897	15372	18893	8179	15745	3045	12134	26636	BERJALAN
38	23254	24852	22183	12405	22485	20577	17654	-29749	16905	6584	14317	6647	BERJALAN
39	23261	12636	25156	7001	24300	21957	16394	7660	22769	25625	15740	20924	BERJALAN
40	24041	2050	23658	10877	25307	5424	16237	10820	18475	13240	21547	16267	BERJALAN



A.1.3 PELETAKAN SENSOR DI PAHA DAN PINGGANG

No	tot_acc1 sensor 1 (fitur 1)	tot_gy1 sensor 1 (fitur2)	tot_acc2 sensor 1 (fitur 3)	tot_gy2 sensor 1 (fitur4)	tot_acc3 sensor 1 (fitur 5)	tot_gy3 sensor 1 (fitur6)	tot_acc1 sensor 2 (fitur 7)	tot_gy1 sensor 2 (fitur 8)	tot_acc2 sensor 2 (fitur 9)	tot_gy2 sensor 2 (fitur10)	tot_acc3 sensor 2 (fitur11)	tot_gy3 sensor 2 (fitur12)	Kelas Klasifikasi
1	25751	6746	25517	1412	25069	8188	16679	194	16597	402	16660	525	BERDIRI
2	25388	3363	25119	9947	25761	8395	16742	6680	15956	6826	16864	1960	BERDIRI
3	25623	10767	25454	3580	25616	838	16397	1015	16617	1075	16543	751	BERDIRI
4	25811	1024	25424	1347	25528	730	16480	632	16517	367	16657	599	BERDIRI
5	25538	303	25680	1473	25639	2293	16644	221	16702	656	16624	422	BERDIRI
6	25481	464	25435	285	25387	653	16545	310	16600	85	16600	265	BERDIRI
7	25777	3732	25746	4167	24818	18387	16753	1380	16701	1822	16652	4624	BERDIRI
8	24935	10233	25604	11952	25322	3268	15768	3856	16773	1296	16842	2568	BERDIRI
9	25708	6115	25598	5160	25549	2671	16336	1592	16574	460	16870	644	BERDIRI
10	25947	10119	25795	3716	25057	16275	15510	11080	16710	3176	14601	12903	BERDIRI
11	23205	422	23279	434	23319	386	16532	361	16692	382	16566	238	DUDUK
12	23244	1393	23951	2247	23494	3252	16699	2106	16340	1063	16425	890	DUDUK
13	23365	369	23433	440	23441	451	16505	245	16527	221	16518	329	DUDUK
14	22982	748	23086	341	23044	537	16603	1200	16445	615	16435	447	DUDUK
15	23594	1455	24166	8132	23490	868	16822	3115	16176	5092	16614	715	DUDUK
16	25543	2600	26407	5519	26329	4670	17012	1537	17292	1183	17314	317	DUDUK
17	26206	6719	26574	1137	26241	350	17789	1795	17728	613	17610	701	DUDUK
18	26748	1050	26792	1150	26375	437	17311	266	17305	284	17434	378	DUDUK
19	25910	320	25748	509	25713	415	17421	265	17427	242	17432	405	DUDUK
20	25311	427	25193	464	25152	477	17275	235	17306	341	17200	3428	DUDUK
21	21907	455	21700	220	21948	910	17619	345	17631	1119	17481	1060	BERBARING
22	21757	988	22068	551	22203	455	17922	743	17953	195	17664	458	BERBARING
23	22164	734	22244	444	22169	477	17561	652	17672	583	17579	278	BERBARING
24	22212	488	22187	584	22268	320	17697	357	17717	617	17728	717	BERBARING
25	21984	551	22096	613	22028	343	17662	664	17887	1326	17521	355	BERBARING
26	22244	458	22137	546	22091	683	17819	483	17550	375	17408	821	BERBARING
27	23530	1546	23443	2331	23030	1388	16793	1393	18016	508	17746	269	BERBARING
28	23503	420	23464	390	23505	329	17754	370	17569	856	17755	352	BERBARING
29	23602	498	23539	540	23564	367	17566	1179	17508	310	17555	216	BERBARING

30	22092	366	22261	455	22191	403	17664	679	17965	503	17674	194	BERBARING
31	26739	7451	26058	7333	26314	8955	12276	-32321	19919	26456	14217	30174	BERJALAN
32	26668	1747	26638	8263	25966	7773	18724	15040	15263	4991	16843	26214	BERJALAN
33	26766	8797	27076	2058	26231	8295	24067	20975	20835	20641	14709	10359	BERJALAN
34	26240	3874	26582	4658	25875	3266	16434	28755	18054	13431	16048	20214	BERJALAN
35	26463	8088	26166	12404	29743	6123	19819	4988	16003	3132	16687	12846	BERJALAN
36	27079	825	27378	13081	26281	6815	14392	3499	15817	2460	18279	12819	BERJALAN
37	26716	5106	26419	3592	-32322	9995	9989	-30952	16276	12359	16614	22792	BERJALAN
38	25812	7687	29893	7350	26162	4599	15393	3188	16234	14703	16757	9064	BERJALAN
39	27278	4886	29097	6427	26029	10102	15353	31765	15655	25946	17623	4800	BERJALAN
40	25970	8568	27941	3762	25701	1204	18848	18685	17978	10182	16577	7173	BERJALAN



A.1.4 PELETAKAN SENSOR DI PAHA DAN KEPALA

No	tot_acc1 sensor 1 (fitur 1)	tot_gy1 sensor 1 (fitur2)	tot_acc2 sensor 1 (fitur 3)	tot_gy2 sensor 1 (fitur4)	tot_acc3 sensor 1 (fitur 5)	tot_gy3 sensor 1 (fitur6)	tot_acc1 sensor 2 (fitur 7)	tot_gy1 sensor 2 (fitur 8)	tot_acc2 sensor 2 (fitur 9)	tot_gy2 sensor 2 (fitur10)	tot_acc3 sensor 2 (fitur11)	tot_gy3 sensor 2 (fitur12)	Kelas Klasifikasi
1	25176	7779	24859	2367	24582	2185	19073	7894	15925	7403	17479	11760	BERDIRI
2	25033	308	25016	1337	24910	308	17096	593	17039	345	17059	338	BERDIRI
3	24700	931	26146	7000	25140	2410	16804	328	17000	1082	17157	737	BERDIRI
4	24859	2743	24333	1051	24699	931	18048	929	16459	288	16748	434	BERDIRI
5	24947	77	25031	506	24913	2674	16876	1282	16857	480	17117	211	BERDIRI
6	25848	8086	26084	3331	25851	1264	14723	9435	16377	3303	17648	1257	BERDIRI
7	24920	4842	25411	23231	25108	14978	17679	638	17308	340	17660	257	BERDIRI
8	25360	1193	25149	1143	25158	1462	17453	372	17527	485	17400	747	BERDIRI
9	25212	439	25192	343	25189	1092	17378	201	17345	365	17388	267	BERDIRI
10	25275	632	25154	577	25227	376	17521	225	17540	485	17681	560	BERDIRI
11	17135	1310	17232	331	17048	766	17142	129	17232	354	17351	206	DUDUK
12	17030	483	17075	431	17212	430	17291	266	17312	222	17238	269	DUDUK
13	17060	634	17047	541	16982	662	17307	3055	17714	480	17480	810	DUDUK
14	16747	2050	16973	425	17154	558	17445	55	17384	131	17471	246	DUDUK
15	17188	448	16971	643	17004	879	17518	566	17229	814	17424	431	DUDUK
16	23258	2645	26115	1645	24565	724	16197	11480	16674	1325	17009	792	DUDUK
17	24295	2447	24492	870	25094	1557	16806	308	16765	268,	17695	5186	DUDUK
18	24486	943	24685	1945	24275	2122	16498	611	16891	499	16949	203	DUDUK
19	24240	542	24158	804	24241	595	16700	362	16975	367	16976	314	DUDUK
20	24746	2572	24889	410	24170	396	16793	353	16293	4974	18974	1429	DUDUK
21	10337	7853	8111	2887	6433	4546	14913	9241	17211	713	17679	2382	BERBARING
22	10167	621	10472	674	10638	1333	17730	872	17653	1742	17425	211	BERBARING
23	11620	955	11126	664	10395	2335	17277	1251	17389	871	17458	437	BERBARING
24	11244	378	10858	2891	12238	504	17449	583	17347	703	17416	365	BERBARING
25	11331	662	11737	358	11504	373	17369	273	17595	169	17169	247	BERBARING
26	11584	351	11533	209	11606	382	17496	408	17482	923	17439	417	BERBARING
27	11594	339	11529	571	11808	511	17344	279	17600	320	17420	484	BERBARING
28	11391	619	11716	634	11770	327	17336	282	17605	225	17283	292	BERBARING
29	11687	425	11661	447	11798	468	17421	452	17291	1101	17501	5481	BERBARING

30	12663	746	12416	7032	10468	6434	17240	306	16983	206	17131	454	BERBARING
31	22869	7852	24110	4296	23090	2689	19192	-14844	16873	26032	15434	20725	BERJALAN
32	24876	7824	25011	8166	23643	1858	15827	28571	16671	11490	12385	24387	BERJALAN
33	23706	4359	21756	911	25111	1820	17073	5293	14691	7447	16679	26040	BERJALAN
34	23832	4257	25641	6988	25949	2940	18110	20591	18871	4065	17962	8233	BERJALAN
35	23075	3012	23326	2878	25459	2866	16685	23308	16194	15541	21039	24201	BERJALAN
36	25694	3424	23881	2188	24341	1864	14213	-32693	17520	25378	13990	-32316	BERJALAN
37	23557	4223	21695	4322	21770	6999	19265	26985	20236	-21191	17604	19266	BERJALAN
38	25459	9001	27295	4141	25807	4031	16311	9800	19674	-16064	11931	10810	BERJALAN
39	26885	1699	22658	6261	24724	10781	13100	29118	13277	27716	11955	-24429	BERJALAN
40	24048	7978	24629	6198	22977	13078	15533	27686	16489	-31423	13109	-28513	BERJALAN



A.1.5 PELETAKAN SENSOR DI KAKI BAGIAN BAWAH DAN LENGAN

No	tot_acc1 sensor 1 (fitur 1)	tot_gy1 sensor 1 (fitur2)	tot_acc2 sensor 1 (fitur 3)	tot_gy2 sensor 1 (fitur4)	tot_acc3 sensor 1 (fitur 5)	tot_gy3 sensor 1 (fitur6)	tot_acc1 sensor 2 (fitur 7)	tot_gy1 sensor 2 (fitur 8)	tot_acc2 sensor 2 (fitur 9)	tot_gy2 sensor 2 (fitur10)	tot_acc3 sensor 2 (fitur11)	tot_gy3 sensor 2 (fitur12)	Kelas Klasifikasi
1	24044	356	24103	330	24151	598	15963	15404	16002	1186	17064	11063	BERDIRI
2	26289	383	25718	959	26076	774	15813	355	15740	522	15810	313	BERDIRI
3	26059	772	25924	630	26032	441	15763	371	15718	228	15668	1318	BERDIRI
4	26108	327	25971	327	25903	446	15830	314	15664	183	15691	470	BERDIRI
5	26057	3512	26146	1040	25733	1681	15703	13622	15649	733	15955	1121	BERDIRI
6	26129	1417	26020	562	26109	856	15767	2577	15685	941	15785	340	BERDIRI
7	26058	400	26078	674	25972	1646	16099	872	16119	425	15793	186	BERDIRI
8	25975	371	26153	449	25839	699	15770	803	15618	498	15608	460	BERDIRI
9	26048	423	26000	474	25921	545	15759	202	15864	1001	15705	798	BERDIRI
10	26001	370	25996	352	25932	611	15752	548	15719	1238	15710	273	BERDIRI
11	23494	6835	26770	9201	25962	427	13796	11276	18180	28721	17187	9305	DUDUK
12	25212	6866	25302	3543	25857	6252	15687	9653	15518	1418	15717	3730	DUDUK
13	25274	434	25285	205	25642	332	15705	777	15669	587	15737	1169	DUDUK
14	24697	534	24459	636	24673	391	15772	451	15744	235	15722	463	DUDUK
15	24226	823	24421	435	24820	1260	15810	220	15802	1548	15887	613	DUDUK
16	26070	539	25985	411	25861	515	15779	969	15863	200	15616	403	DUDUK
17	25941	789	25882	655	25797	427	15625	4755	15336	5036	15997	944	DUDUK
18	26449	1788	26027	772	26040	1159	15403	8918	17614	4303	15849	1155	DUDUK
19	26014	966	26090	499	26133	497	15733	555	15735	1174	15837	814	DUDUK
20	26083	417	26110	1159	26053	502	15776	793	15985	458	15755	523	DUDUK
21	26925	8250	25405	3645	26947	7736	17651	16122	16190	15223	16440	5397	BERBARING
22	25778	2224	25962	1403	25840	677	18218	6191	16586	3090	16866	2555	BERBARING
23	26130	413	26191	391	26183	434	16639	181	16685	192	16685	511	BERBARING
24	26057	1251	26159	755	26134	900	16687	2436	16665	1464	16808	1117	BERBARING
25	26465	395	26524	430	26444	447	16871	340	17574	11589	17239	14153	BERBARING
26	26518	315	26428	309	26513	454	17584	4441	17908	7180	17539	13746	BERBARING
27	26505	402	26453	445	26515	383	17215	291	17347	286	17200	345	BERBARING
28	26396	443	26427	417	26472	432	17181	1483	17097	769	17230	391	BERBARING
29	26454	397	26343	407	26414	474	17072	816	17171	429	17132	252	BERBARING

30	26371	400	26573	374	26448	429	17105	385	17005	222	17133	295	BERBARING
31	27024	-20934	25357	13529	26773	-23628	14993	11718	16107	24738	14927	7937	BERJALAN
32	32173	-26505	22906	19863	24050	2506	15117	19879	15591	-29775	17756	24437	BERJALAN
33	23367	29791	26700	6634	26261	17228	17625	25935	16930	25878	17487	-22023	BERJALAN
34	21928	18408	27263	15445	23011	22814	16754	-32293	22418	-26289	16658	-19470	BERJALAN
35	22778	-22134	24205	-28779	24632	-25117	20071	-24898	15760	11144	18191	18114	BERJALAN
36	24088	11632	23273	31932	25616	15776	19076	1320	14980	26844	15388	14553	BERJALAN
37	23496	-19430	29135	-20957	24335	-29497	15756	6280	16004	12420	15801	23491	BERJALAN
38	24470	9948	23596	-15943	22817	5622	16152	-27828	16086	20251	15505	8480	BERJALAN
39	27583	15229	32164	-12814	25066	-17035	14545	-16868	15863	32644	16412	7577	BERJALAN
40	28745	20757	23672	22368	25888	-18250	15974	11287	16130	14121	15730	8049	BERJALAN



A.1.6 PELETAKAN SENSOR DI KAKI BAGIAN BAWAH DAN PINGGANG

No	tot_acc1 sensor 1 (fitur 1)	tot_gy1 sensor 1 (fitur2)	tot_acc2 sensor 1 (fitur 3)	tot_gy2 sensor 1 (fitur4)	tot_acc3 sensor 1 (fitur 5)	tot_gy3 sensor 1 (fitur6)	tot_acc1 sensor 2 (fitur 7)	tot_gy1 sensor 2 (fitur 8)	tot_acc2 sensor 2 (fitur 9)	tot_gy2 sensor 2 (fitur10)	tot_acc3 sensor 2 (fitur11)	tot_gy3 sensor 2 (fitur12)	Kelas Klasifikasi
1	24930	418	24804	458	24915	1095	16264	376	16258	245	16168	337	BERDIRI
2	24513	3574	24764	1709	24686	1712	16459	726	16300	1354	16158	1188	BERDIRI
3	24957	9065	24406	4334	24733	1767	16273	317	16240	373	16352	586	BERDIRI
4	24842	1402	24910	1644	24737	1695	16217	2461	16221	1724	16287	2124	BERDIRI
5	24767	1370	24823	2005	24499	3663	16394	7020	16852	2933	16895	883	BERDIRI
6	24673	1380	24462	1264	24667	5318	16611	3247	16308	962	16390	608	BERDIRI
7	24826	3288	24717	263	24664	406	16438	366	16308	125	16443	277	BERDIRI
8	24617	1171	24977	908	24677	2040	16404	335	16333	533	16232	183	BERDIRI
9	24808	556	24684	208	24671	404	16588	472	16402	678	16434	552	BERDIRI
10	24663	3057	24667	829	24581	1208	16584	1028	16532	1364	16410	457	BERDIRI
11	25716	565	25697	443	25675	466	16343	344	16310	344	16399	209	DUDUK
12	25802	422	25766	404	25776	388	16373	478	16467	278	16326	322	DUDUK
13	25841	644	25294	6956	26526	3786	16212	2865	17007	5401	16369	3746	DUDUK
14	25448	2472	25853	5841	25755	3231	16400	1329	16204	1638	16410	841	DUDUK
15	25721	343	25692	539	25743	480	16332	296	16397	345	16381	240	DUDUK
16	26481	396	26365	455	26314	337	15533	261	15568	599	15689	1138	DUDUK
17	26097	530	26337	617	26405	453	15511	712	15542	705	15650	395	DUDUK
18	26353	511	26403	463	26364	442	15676	204	15643	397	15502	282	DUDUK
19	26371	596	26411	438	26312	442	15549	182	15651	205	15649	228	DUDUK
20	26953	1056	26587	2960	25974	2800	15201	17014	15594	7877	15467	7201	DUDUK
21	23118	435	22956	439	22903	396	17582	313	17507	296	17590	222	BERBARING
22	22956	680	22983	370	22999	414	17593	368	17630	174	17364	792	BERBARING
23	23058	509	22961	412	23060	451	17576	370	17545	275	17840	241	BERBARING
24	22900	396	22926	471	23001	643	17485	233	17472	380	17559	207	BERBARING
25	23022	421	22966	386	22990	411	17659	228	17551	302	17522	315	BERBARING
26	23083	484	22996	482	23081	414	17460	277	17373	137	17484	225	BERBARING
27	23084	520	23093	376	23042	410	17499	232	17541	225	17518	270	BERBARING
28	22964	486	22993	495	22980	612	17579	246	17487	260	17363	774	BERBARING
29	22820	504	23093	452	22972	431	17543	265	17546	231	17623	250	BERBARING

30	22964	344	23041	372	23096	483	17565	243	17490	444	17544	305	BERBARING
31	25649	8518	25565	7710	26995	4612	12929	32279	15798	-32729	21010	-18031	BERJALAN
32	30276	15542	26035	4878	27083	2446	14952	13745	12250	-24090	18628	-9646	BERJALAN
33	25356	10363	-32103	20173	26546	3592	28524	-32519	20772	-13427	13816	-30357	BERJALAN
34	26992	14356	27080	13565	27567	7037	11916	-11907	15571	15990	14618	-10626	BERJALAN
35	27658	4594	26224	5160	26061	4838	14840	19496	24829	-26428	12662	26061	BERJALAN
36	25741	4461	25984	2942	28048	1715	14198	-25321	15852	17162	13820	28858	BERJALAN
37	31607	4812	26223	13257	27963	10437	26309	27130	13018	-31953	16356	-30753	BERJALAN
38	25695	2698	25559	3504	25984	5413	19449	15679	17199	30724	16745	14917	BERJALAN
39	24861	8088	25652	3578	25888	4888	18667	25935	12012	12063	15886	29428	BERJALAN
40	26335	8267	28384	3257	29038	10577	18246	23962	13739	-32279	16977	-28533	BERJALAN



A.1.7 PELETAKAN SENSOR DI KAKI BAGIAN BAWAH DAN KEPALA

No	tot_acc1 sensor 1 (fitur 1)	tot_gy1 sensor 1 (fitur2)	tot_acc2 sensor 1 (fitur 3)	tot_gy2 sensor 1 (fitur4)	tot_acc3 sensor 1 (fitur 5)	tot_gy3 sensor 1 (fitur6)	tot_acc1 sensor 2 (fitur 7)	tot_gy1 sensor 2 (fitur 8)	tot_acc2 sensor 2 (fitur 9)	tot_gy2 sensor 2 (fitur10)	tot_acc3 sensor 2 (fitur11)	tot_gy3 sensor 2 (fitur12)	Kelas Klasifikasi
1	24902	14168	23821	22422	27923	14231	17538	471	17340	576	17942	3229	BERDIRI
2	26218	889	26183	973	26241	345	17706	9609	17181	5049	17541	3830	BERDIRI
3	26005	322	26078	647	25947	725	17243	748	17270	825	17736	1528	BERDIRI
4	25753	987	25947	773	26093	470	17355	752	17633	598	17433	717	BERDIRI
5	25910	1094	26012	365	26028	305	17554	927	17684	412	17274	521	BERDIRI
6	26410	3502	26178	510	26307	741	17671	2840	17476	769	17268	1181	BERDIRI
7	26334	1247	26436	552	26528	423	17081	501	17096	311	17170	657	BERDIRI
8	26396	3865	26524	1009	26453	1503	17057	197	17098	312	17159	144	BERDIRI
9	26459	364	26523	398	26430	409	17557	2521	17853	1962	18716	7259	BERDIRI
10	26442	475	26498	384	26421	392	21122	1564	14252	2200	18437	1406	BERDIRI
11	24567	2012	25092	1088	24464	1066	15491	2276	16453	-19765	15281	7369	DUDUK
12	24420	301	24570	416	24685	323	15739	324	15805	291	15736	686	DUDUK
13	24660	374	23687	2747	23098	1448	15774	510	15685	1013	16030	4041	DUDUK
14	24577	324	24535	338	24383	1068	15731	512	15737	276	15713	504	DUDUK
15	24082	2796	24160	1136	24017	290	15925	4141	15591	1663	16058	1068	DUDUK
16	28951	8779	25691	2114	24071	7268	17280	783	17946	1354	17768	1049	DUDUK
17	24649	839	24658	454	24613	1846	17378	411	17342	269	17264	231	DUDUK
18	24631	374	24587	344	24648	464	17291	393	17465	230	17443	249	DUDUK
19	24608	295	24618	1143	24710	960	17428	336	17312	325	17239	492	DUDUK
20	24526	668	24646	337	24608	802	17222	329	17219	287	17124	317	DUDUK
21	17662	4446	13069	11143	12008	3722	31661	26007	21537	3503	16081	589	BERBARING
22	11706	740	11885	884	12008	2252	19095	7520	17913	3092	16672	12611	BERBARING
23	12882	479	12936	361	12961	342	16142	5033	16152	4341	16472	198	BERBARING
24	12955	468	12968	1077	12943	683	15279	29922	15804	6135	16973	3820	BERBARING
25	12356	633	12143	509	12097	537	17677	22196	15880	3211	15954	2893	BERBARING
26	11929	386	11979	539	11935	1929	19705	27278	21476	25947	16296	6879	BERBARING
27	12123	345	12307	455	12386	390	17361	466	17363	301	16274	1341	BERBARING
28	23744	368	25976	1809	23816	1207	19166	12450	15522	10860	18606	2617	BERBARING
29	25572	1000	25411	724	25261	573	16592	153	16560	542	16368	206	BERBARING

30	25381	478	25356	457	25434	519	16728	231	16606	379	16562	545	BERBARING
31	27050	3433	24469	2581	27573	1243	14488	19465	15991	20149	16226	7797	BERJALAN
32	25678	4493	24613	3041	25908	3417	12006	-25427	11066	11978	12964	31277	BERJALAN
33	25328	1230	26750	3855	24284	11420	12438	-22051	12190	-19354	13998	-20304	BERJALAN
34	23523	1274	24254	3102	24141	2285	16843	31492	19163	8298	18593	29995	BERJALAN
35	24965	2891	23774	4339	23604	5659	16588	24610	17404	25702	22840	29386	BERJALAN
36	22121	21782	25441	12528	24282	17909	15003	-16647	11200	-26016	10266	19710	BERJALAN
37	23148	8716	21612	10525	23539	5555	12229	23682	19357	10606	21067	26221	BERJALAN
38	23261	15963	24035	3409	25114	3896	29132	-10733	14386	15724	13583	30355	BERJALAN
39	24888	3766	23237	2246	22310	13168	19854	26418	22346	12318	10316	-20056	BERJALAN
40	23811	5906	23018	2564	21607	2913	18294	30405	14959	-18676	19611	-19126	BERJALAN



A.1.8 PELETAKAN SENSOR DI LENGAN DAN PINGGANG

No	tot_acc1 sensor 1 (fitur 1)	tot_gy1 sensor 1 (fitur2)	tot_acc2 sensor 1 (fitur 3)	tot_gy2 sensor 1 (fitur4)	tot_acc3 sensor 1 (fitur 5)	tot_gy3 sensor 1 (fitur6)	tot_acc1 sensor 2 (fitur 7)	tot_gy1 sensor 2 (fitur 8)	tot_acc2 sensor 2 (fitur 9)	tot_gy2 sensor 2 (fitur10)	tot_acc3 sensor 2 (fitur11)	tot_gy3 sensor 2 (fitur12)	Kelas Klasifikasi
1	26837	18251	25804	5179	25650	6752	16007	1390	16014	1589	19366	12716	BERDIRI
2	21791	-20543	28577	6840	22829	9599	16564	9132	17289	1639	14309	18065	BERDIRI
3	25859	9184	25651	546	25756	1870	16217	670	16065	2206	16168	1240	BERDIRI
4	25316	3289	25415	960	25428	414	16043	675	16196	2078	17734	5097	BERDIRI
5	25396	150	25494	733	25371	1916	15982	331	16072	265	16056	465	BERDIRI
6	25627	1536	25475	924	25503	3263	15942	864	16016	284	16183	467	BERDIRI
7	25217	1018	25330	1275	26007	4248	16117	342	16025	702	16200	1742	BERDIRI
8	25192	155	25216	1218	25717	2666	16267	526	16122	322	16145	789	BERDIRI
9	25137	1163	25203	529	25368	1349	16081	300	16055	386	16134	265	BERDIRI
10	25371	903	25206	2386	25334	1138	15933	2180	15915	363	16141	662	BERDIRI
11	26414	384	26579	364	26612	351	15850	1017	15681	790	15794	600	DUDUK
12	26406	409	26390	364	26370	428	15851	1638	15838	1005	16557	467	DUDUK
13	26445	324	26388	476	26377	602	15904	354	15918	1205	15545	580	DUDUK
14	26493	478	26466	357	26421	1695	15986	1531	15874	3481	16017	4956	DUDUK
15	26407	476	26498	565	26498	638	15783	1158	16184	1121	18328	13283	DUDUK
16	25935	4754	25795	377	26428	4331	16088	518	16331	407	16354	530	DUDUK
17	26003	632	26016	347	25980	378	16435	345	16527	373	16394	271	DUDUK
18	25823	978	25960	1643	26057	405	16618	5639	16678	4132	16454	489	DUDUK
19	25767	275	25710	394	25906	465	16330	278	16464	239	16414	197	DUDUK
20	25960	459	25828	680	25794	2274	16343	303	16419	275	16305	318	DUDUK
21	17151	460	17289	214	17057	419	14839	3902	15608	2701	15691	289	BERBARING
22	17291	700	17410	491	17307	434	15883	225	15888	195	15772	321	BERBARING
23	17249	460	17572	618	17471	350	15756	260	15773	237	15824	247	BERBARING
24	17274	603	17459	469	17385	392	15834	204	15760	253	15888	228	BERBARING
25	17490	439	17635	445	17552	278	15796	297	15809	342	15938	277	BERBARING
26	17494	663,	17645	511	17558	312	15930	272	15877	258	15865	273	BERBARING
27	17573	721	17572	667	17547	519	15848	315	15926	203	15883	259	BERBARING
28	17512	563	17646	314	17473	356	15803	270	15815	300	15617	336	BERBARING
29	17557	569	17511	450	17795	315	15802	234	15874	251	15881	230	BERBARING

30	17557	569	17511	450	17795	315	15669	264	15946	229	15784	235	BERBARING
31	27565	1398	27044	7170	27397	13802	16457	8572	16654	4854	16215	-31329	BERJALAN
32	25053	8656	25953	32298	25718	8745	16214	23593	12997	11462	13132	11194	BERJALAN
33	26193	7106	31397	4350	32307	7603	16214	23593	12997	11462	13132	11194	BERJALAN
34	28299	12391	26964	4902	26670	3294	15006	-25672	15717	31867	14794	7826	BERJALAN
35	25459	2558	26289	6326	23604	8690	15589	16482	15996	3952	16724	24131	BERJALAN
36	26475	3774	32621	9726	26177	1134	17018	23683	18048	23821	16380	4694	BERJALAN
37	25709	5050	26228	13036	30098	5340	13474	22329	16138	17920	17029	9342	BERJALAN
38	25873	3165	28366	6639	25664	5239	15667	2796	15394	22292	15717	10226	BERJALAN
39	25818	21018	25213	8749	25225	2887	16509	29868	15821	17138	16126	17075	BERJALAN
40	28126	2564	23385	9149	25938	15241	14758	8865	14092	21181	16857	-23719	BERJALAN



A.1.9 PELETAKAN SENSOR DI LENGAN DAN KEPALA

No	tot_acc1 sensor 1 (fitur 1)	tot_gy1 sensor 1 (fitur2)	tot_acc2 sensor 1 (fitur 3)	tot_gy2 sensor 1 (fitur4)	tot_acc3 sensor 1 (fitur 5)	tot_gy3 sensor 1 (fitur6)	tot_acc1 sensor 2 (fitur 7)	tot_gy1 sensor 2 (fitur 8)	tot_acc2 sensor 2 (fitur 9)	tot_gy2 sensor 2 (fitur10)	tot_acc3 sensor 2 (fitur11)	tot_gy3 sensor 2 (fitur12)	Kelas Klasifikasi
1	25357	16450	26093	9090	25266	11276	17252	2494	20587	4972	17638	3340	BERDIRI
2	25008	4988	25179	549	24938	770	17661	4778	17467	913	17124	1464	BERDIRI
3	26391	8190	26402	791	26545	3961	16813	680	17091	406	16876	861	BERDIRI
4	24862	794	24999	647	24921	307	17086	287	16967	129	17127	275	BERDIRI
5	25022	3634	25410	748	25134	1128	16840	509	17076	677	17130	694	BERDIRI
6	25072	904	24958	1715	25175	1053	17144	1361	17073	727	17186	649	BERDIRI
7	25125	430	25279	358	25248	363	17027	468	17276	471	17063	322	BERDIRI
8	25225	430	24958	466	25067	882	17157	495	17129	493	17079	169	BERDIRI
9	25141	772	25128	1461	25047	612	17289	416	17071	150	17158	675	BERDIRI
10	23480	523	23767	629	23683	1406	16061	695	15985	750	16073	835	BERDIRI
11	24170	594	24142	377	24322	446	16091	675	16055	514	16088	276	DUDUK
12	24079	181	24062	272	24264	337	16089	651	16131	635	16019	540	DUDUK
13	24484	349	24436	345	24409	330	15868	369	15972	755	15962	312	DUDUK
14	24521	364	24317	307	24402	594	15680	420	15830	754	16027	594	DUDUK
15	25859	398	25819	470	25748	1640	17369	397	17299	572	17198	433	DUDUK
16	25776	1008	25920	1445	25884	1312	17210	506	17194	390	17245	102	DUDUK
17	25770	1014	25841	1149	25778	858	17274	232	17253	212	17291	278	DUDUK
18	25889	501	25929	608	26139	283	17289	693	17217	664	17265	423	DUDUK
19	25849	990	25694	457	25695	439	17289	693	17217	664	17265	423	DUDUK
20	20276	923	19802	443	19915	434	16409	660	16374	168	16439	415	DUDUK
21	20144	578	19959	595	19899	670	16338	296	16439	243	16323	193	BERBARING
22	19941	400	19683	693	19937	384	16262	199	16262	1583	16431	323	BERBARING
23	20027	478	20203	573	20470	979	16116	179	16304	716	16156	353	BERBARING
24	20030	515	19851	218	20217	351	16377	292	16255	256	16461	360	BERBARING
25	19743	15694	19969	2845	21142	3895	17138	822	16677	335	15537	3339	BERBARING
26	20421	9950	20872	647	20559	721	16646	126	16601	269	16642	310	BERBARING
27	21032	2899	21263	817	21135	537	16572	126	16612	263	16418	287	BERBARING
28	20644	3182	20890	2952	21526	294	16485	316	16462	256	16365	218	BERBARING
29	21670	400	21589	311	21547	404	16644	312	16505	108	16568	413	BERBARING

30	25676	2119	19885	3937	24367	13491	15898	-30404	11755	30971	15153	31786	BERBARING
31	25294	2020	23171	10854	24585	6914	20509	-23605	14736	5477	22192	-15065	BERJALAN
32	24651	5950	24567	2173	25252	3787	21562	32062	26021	18721	18591	-24449	BERJALAN
33	23614	6444	23280	2676	26881	1050	22554	21847	14773	9159	16438	11607	BERJALAN
34	22621	14128	23668	2031	26921	9724	14798	23248	14893	29272	12998	28564	BERJALAN
35	23167	-14992	26279	11551	21916	16567	15914	20702	18168	6899	18198	3391	BERJALAN
36	27989	5910	24873	21745	24733	16852	15740	2713	18472	2444	17552	844	BERJALAN
37	24554	7747	26295	7561	25685	31724	17081	2879	18101	1625	16924	4568	BERJALAN
38	24354	-31611	24952	29170	20925	23385	17514	5364	16093	5747	17345	3132	BERJALAN
39	29890	-17383	23884	-12866	26489	-15765	17458	5905	22641	5905	17314	5442	BERJALAN
40	22622	14129	23600	141	26967	9624	16798	22248	15893	29472	13998	23564	BERJALAN



A.1.10 PELETAKAN SENSOR DI PINGGANG DAN KEPALA

No	tot_acc1 sensor 1 (fitur 1)	tot_gy1 sensor 1 (fitur2)	tot_acc2 sensor 1 (fitur 3)	tot_gy2 sensor 1 (fitur4)	tot_acc3 sensor 1 (fitur 5)	tot_gy3 sensor 1 (fitur6)	tot_acc1 sensor 2 (fitur 7)	tot_gy1 sensor 2 (fitur 8)	tot_acc2 sensor 2 (fitur 9)	tot_gy2 sensor 2 (fitur10)	tot_acc3 sensor 2 (fitur11)	tot_gy3 sensor 2 (fitur12)	Kelas Klasifikasi
1	25354	6896	26152	3073	26990	2111	18605	3038	17684	1323	17422	1823	BERDIRI
2	26182	654	26218	409	26442	1255	17359	802	17612	207	17689	964	BERDIRI
3	26238	1309	26308	445	26229	651	17553	526	17513	431	17445	589	BERDIRI
4	26213	715	26146	1205	26313	797	17419	479	17376	513	17389	1322	BERDIRI
5	26385	452	26286	422	26447	613	17245	593	17252	571	17269	259	BERDIRI
6	26326	294	26216	758	26128	571	17288	580	17199	961	17367	628	BERDIRI
7	26289	449	26359	184	26265	701	17300	673	17209	162	17303	237	BERDIRI
8	26204	826	26213	993	26281	456	17194	840	17534	1415	17366	347	BERDIRI
9	26364	366	26238	619	26363	446	17305	943	17177	429	17233	704	BERDIRI
10	26229	795	26359	302	26453	480	17404	501	17347	673	17312	479	BERDIRI
11	23929	856	23879	1033	24383	248	16571	1201	16346	838	16507	266	DUDUK
12	23618	1526	24604	879	24346	425	16476	316	16548	588	16557	1325	DUDUK
13	24402	362	24563	597	24247	489	17321	1054	16486	441	16466	476	DUDUK
14	23612	845	23829	913	24558	331	17321	1054	16486	441	16466	476	DUDUK
15	24789	611	24128	656	24688	501	16359	544	16435	739	16776	364	DUDUK
16	24216	902	24376	1147	24368	229	16480	508	16728	246	16341	437	DUDUK
17	26492	552	26381	325	26585	236	17388	9584	17420	3031	18352	1424	DUDUK
18	26264	293	26305	467	26197	754	17194	748	17103	884	17292	436	DUDUK
19	26156	408	26082	540	26099	736	17127	196	17068	346	17487	485	DUDUK
20	26263	301	26163	501	26151	1332	17127	196	17068	346	17487	485	DUDUK
21	18056	264	17890	617	18098	141	16332	329	16126	289	16213	491	BERBARING
22	18015	362	18114	404	18108	346	16235	270	16148	264	16146	206	BERBARING
23	17904	415	18105	390	18165	296	16322	256	16261	228	16192	198	BERBARING
24	17935	394	18012	453	18138	360	6102	388	16156	289	16276	242	BERBARING
25	17749	15752	23047	791	22463	5293	16961	2123	16513	3893	17472	3830	BERBARING
26	12644	2258	12675	1432	12772	1482	19788	2457	18034	1543	17459	2228	BERBARING
27	12701	628	12979	716,	12654	644	17011	268	16771	354	17739	690	BERBARING
28	12772	1330	12846	243	13039	821	17114	1863	17380	1275	16798	867	BERBARING
29	13264	592	13377	599	13184	259	17425	494	16863	662	16524	154	BERBARING

30	13372	643	13363	479	13405	448	16852	383	16847	313	16713	330	BERBARING
31	22328	2753	27893	3174	23230	2800	23057	11041	14033	11479	15929	2476	BERJALAN
32	25528	7124	27439	850	26577	3018	16126	3798	18225	6453	16459	2153	BERJALAN
33	24490	10817	27740	4780	24267	3268	15875	13132	12552	15022	16633	8879	BERJALAN
34	23182	3379	25888	14189	24775	7720	14702	5463	16485	12162	14745	9594	BERJALAN
35	25705	2219	21180	15176	-31752	7251	19522	14645	11851	31281	14560	19635	BERJALAN
36	26719	7293	27208	10475	26261	11621	17509	2183	19333	9662	17719	8255	BERJALAN
37	26506	17055	26709	6210	28683	4672	16943	5604	20641	3269	17224	2772	BERJALAN
38	25684	5056	27430	12019	25520	10625	16595	6807	18558	8939	17465	8971	BERJALAN
39	25823	3470	26225	5964	28633	2134	17417	8442	17650	6811	17341	3114	BERJALAN
40	26600	8463	25701	14042	29833	5218	19539	6021	17892	4193	15640	2709	BERJALAN



LAMPIRAN B DATA UJI

B.1 DATA UJI SISTEM DENGAN SATU SENSOR

B.1.1 PELETAKAN SENSOR DI PAHA

Pengujian	tot_acc1 (fitur 1)	tot_gy1 (fitur2)	tot_acc2 (fitur 3)	tot_gy2 (fitur4)	tot_acc3 (fitur 5)	tot_gy3 (fitur6)	Klasifikasi Sistem	Klasifikasi Sebenarnya	Kesesu- aian
Orang ke- 1	25814	781	25841	1662	25662	1856	BERDIRI	BERDIRI	1
	25789	452	25902	583	25772	657	BERDIRI	BERDIRI	1
	25828	477	25833	445	25741	1046	BERDIRI	BERDIRI	1
	27054	7184	28192	5354	27042	9415	BERDIRI	BERDIRI	1
	26341	1099	26179	1279	26318	1452	BERDIRI	BERDIRI	1
	19521	469	19647	429	19460	691	DUDUK	DUDUK	1
	19196	642	19481	488	19527	616	DUDUK	DUDUK	1
	19378	569	19489	329	19601	2639	DUDUK	DUDUK	1
	19515	596	19655	351	19603	636	DUDUK	DUDUK	1
	19488	469	19541	413	19547	378	DUDUK	DUDUK	1
	20365	465	20431	442	20331	505	BERBARING	BERBARING	1
	20437	375	20368	426	20374	447	BERBARING	BERBARING	1
	20384	417	20423	338	20350	411	BERBARING	BERBARING	1
	20331	493	20364	20201	20201	430	BERBARING	BERBARING	1
	20217	452	20304	477	20336	398	BERBARING	BERBARING	1
	25916	17185	27391	-24793	21952	30205	BERJALAN	BERJALAN	1
	24994	-32697	24598	-31723	22999	-19967	BERJALAN	BERJALAN	1
	24112	17950	23437	27999	-32448	16465	BERJALAN	BERJALAN	1
	25264	-18804	23860	-24178	26150	7420	BERJALAN	BERJALAN	1
	26878	15498	24628	9328	24829	17904	BERJALAN	BERJALAN	1
Orang ke- 2	26407	872	26460	1098	27196	11456	BERDIRI	BERDIRI	1
	25859	811	25741	904	25984	1828	BERDIRI	BERDIRI	1
	25907	419	25981	708	25803	1131	BERDIRI	BERDIRI	1
	25796	267	25954	696	25942	563	BERDIRI	BERDIRI	1
	26035	433	25926	1012	26020	609	BERDIRI	BERDIRI	1
	19560	427	19529	425	19446	431	DUDUK	DUDUK	1
	19513	563	19604	458	19548	507	DUDUK	DUDUK	1
	19643	376	19672	340	19617	437	DUDUK	DUDUK	1
	19507	503	19608	432	19504	666	DUDUK	DUDUK	1
	19739	398	19415	495	19617	445	DUDUK	DUDUK	1
	20373	425	20292	447	20395	428	BERBARING	BERBARING	1
	20383	400	20382	478	20378	512	BERBARING	BERBARING	1
	20327	465	20519	429	20312	490	BERBARING	BERBARING	1
	20318	465	20451	503	20331	477	BERBARING	BERBARING	1
	20345	406	20396	466	20339	401	BERBARING	BERBARING	1
	25319	2117	23670	9083	25731	14323	BERJALAN	BERJALAN	1
	23799	-29042	26195	-29021	25809	10844	BERJALAN	BERJALAN	1
	24851	-21832	25714	23399	24780	3027	BERDIRI	BERJALAN	0
	26830	-16466	28260	10518	24984	6230	BERJALAN	BERJALAN	1
	24780	27592	22921	-18803	23903	-30961	BERJALAN	BERJALAN	1
Orang ke- 3	26085	1798	26082	929	26047	544	BERDIRI	BERDIRI	1
	25413	3296	25253	1027	26072	1166	BERDIRI	BERDIRI	1
	25886	1671	25719	2326	25965	2028	BERDIRI	BERDIRI	1
	27463	11900	23629	19800	24919	5393	BERJALAN	BERDIRI	0
	25111	4706	25174	3889	25244	912	BERDIRI	BERDIRI	1
	19507	407	19349	356	19640	478	DUDUK	DUDUK	1
	19471	502	19521	474	19499	468	DUDUK	DUDUK	1
	19672	516	19601	449	19538	453	DUDUK	DUDUK	1

	19576	417	19504	428	19676	510	DUDUK	DUDUK	1
	19714	441	195658	393	19481	411	DUDUK	DUDUK	1
	20270	451	20319	278	20276	441	BERBARING	BERBARING	1
	20292	450	20501	414	20254	415	BERBARING	BERBARING	1
	20221	487	20374	289	20248	469	BERBARING	BERBARING	1
	20333	400	20369	443	20281	473	BERBARING	BERBARING	1
	20317	401	20242	438	20355	486	BERBARING	BERBARING	1
	27379	4044	26211	-20981	2647	12203	BERJALAN	BERJALAN	1
	23715	3244	24351	2872	25103	2993	BERDIRI	BERJALAN	0
	31264	-30054	28131	6765	23826	14860	BERJALAN	BERJALAN	1
	26670	20933	25581	22845	20043	14587	BERJALAN	BERJALAN	1
	26871	-31877	28593	17204	28611	-30425	BERJALAN	BERJALAN	1
Orang ke-4	25352	603	25314	1160	25243	561	BERDIRI	BERDIRI	1
	25217	4654	25592	1765	25170	5024	BERDIRI	BERDIRI	1
	25506	661	25529	562	25357	558	BERDIRI	BERDIRI	1
	9184	13169	5140	6229	6391	6879	BERDIRI	BERDIRI	1
	9777	1438	9206	2257	10702	2775	BERDIRI	BERDIRI	1
	19743	416	19585	480	19450	406	DUDUK	DUDUK	1
	19634	453	19542	491	19653	448	DUDUK	DUDUK	1
	19486	429	19602	480	19648	513	DUDUK	DUDUK	1
	19600	491	19647	481	19690	564	DUDUK	DUDUK	1
	19509	415	19559	408	19683	421	DUDUK	DUDUK	1
	20177	522	20252	220	20270	578	BERBARING	BERBARING	1
	20277	480	20244	460	20248	362	BERBARING	BERBARING	1
	20158	580	20259	430	20450	484	BERBARING	BERBARING	1
	20318	442	20224	397	20303	370	BERBARING	BERBARING	1
	20323	446	20346	419	20266	410	BERBARING	BERBARING	1
	31230	2528	20667	6522	26007	4688	BERDIRI	BERJALAN	0
	28773	18678	26878	11091	19941	8999	BERJALAN	BERJALAN	1
	25718	7965	25597	2580	28675	31527	BERJALAN	BERJALAN	1
	28397	2360	28945	7311	25526	11311	BERDIRI	BERJALAN	0
	25124	12562	28679	10856	23776	20837	BERJALAN	BERJALAN	1

B.1.2 PELETAKAN SENSOR DI KAKI BAGIAN BAWAH

Pengujian ke-	tot_acc1 (fitur 1)	tot_gy1 (fitur2)	tot_acc2 (fitur 3)	tot_gy2 (fitur4)	tot_acc3 (fitur 5)	tot_gy3 (fitur6)	Klasifikasi Sistem	Klasifikasi Sebenarnya	Kesesuaian
Orang ke-1	24527	8478	21609	11660	24668	5189	BERJALAN	BERDIRI	0
	25056	16364	24687	2756	24909	1789	BERDIRI	BERDIRI	1
	24793	467	24797	343	24916	243	BERDIRI	BERDIRI	1
	24900	633	24905	1186	24862	543	BERDIRI	BERDIRI	1
	25276	2997	25801	11406	26486	15331	BERDIRI	BERDIRI	1
	25815	4680	23665	21824	25520	19769	BERJALAN	DUDUK	0
	25416	6145	25153	4733	25359	4364	BERDIRI	DUDUK	0
	25186	180	25263	598	25297	328	BERDIRI	DUDUK	0
	25297	307	25429	455	25356	197	BERDIRI	DUDUK	0
	25390	858	25218	738	25127	919	BERDIRI	DUDUK	0
	23538	413	23657	455	23505	513	BERBARING	BERBARING	1
	23575	398	23521	429	23558	413	BERBARING	BERBARING	1
	23544	482	23287	751	23479	753	BERBARING	BERBARING	1
	23572	407	23508	438	23504	622	BERBARING	BERBARING	1
	23666	748	21619	13495	24336	9670	BERBARING	BERBARING	1
	22221	3740	21435	14981	29036	-8782	BERJALAN	BERJALAN	1
	22606	-22090	23000	-18843	24624	27666	BERJALAN	BERJALAN	1
	27842	26599	27108	-21657	27837	22664	BERJALAN	BERJALAN	1
	26449	-12899	25417	25132	29799	-32759	BERJALAN	BERJALAN	1
	25518	5312	18139	-31934	28785	-17382	BERJALAN	BERJALAN	1
Orang ke-2	25335	5622	25109	2379	25227	923	BERDIRI	BERDIRI	1
	25359	1185	25454	1692	25437	2289	BERDIRI	BERDIRI	1
	25449	3633	25505	4109	25459	689	BERDIRI	BERDIRI	1
	25257	2622	25066	1625	25246	583	BERDIRI	BERDIRI	1
	25028	463	24939	667	25096	374	BERDIRI	BERDIRI	1
	25394	611	26403	4210	26696	6400	BERDIRI	DUDUK	0
	24192	2681	24965	2849	25349	1136	BERDIRI	DUDUK	0
	25243	1058	25134	1127	25187	476	BERDIRI	DUDUK	0
	25256	297	25325	503	25433	156	BERDIRI	DUDUK	0
	25340	769	25203	1218	25290	907	BERDIRI	DUDUK	0
	22282	1077	22437	278	22538	409	BERBARING	BERBARING	1
	22371	481	22537	415	22373	411	BERBARING	BERBARING	1
	22521	405	22424	486	22603	431	BERBARING	BERBARING	1
	22486	842	22380	1114	22553	484	BERBARING	BERBARING	1
	22532	415	22574	520	22457	429	BERBARING	BERBARING	1
	26796	30371	32501	-19161	30083	-31305	BERJALAN	BERJALAN	1
	24049	-32178	20359	26132	24703	16392	BERJALAN	BERJALAN	1
	28018	-21765	22499	16917	24605	8945	BERJALAN	BERJALAN	1
	19693	25621	26746	-29680	20327	-28842	BERJALAN	BERJALAN	1
	25023	32549	24615	-27588	25142	-23252	BERJALAN	BERJALAN	1
Orang ke-3	25386	754	25581	465	25547	1988	BERDIRI	BERDIRI	1
	25591	408	25430	640	25583	657	BERDIRI	BERDIRI	1
	25812	557	25540	405	25439	421	BERDIRI	BERDIRI	1
	25528	843	25534	590	25332	418	BERDIRI	BERDIRI	1
	25542	2472	25432	2217	24950	445	BERDIRI	BERDIRI	1
	25435	394	25346	480	25255	1272	BERDIRI	DUDUK	0
	25290	246	25163	679	25140	519	BERDIRI	DUDUK	0
	25288	1253	25327	1745	25146	422	BERDIRI	DUDUK	0
	25257	538	25092	474	25334	541	BERDIRI	DUDUK	0
	25299	322	25284	318	25198	510	BERDIRI	DUDUK	0
	22439	417	22486	488	22496	466	BERBARING	BERBARING	1
	22622	374	22541	378	22582	447	BERBARING	BERBARING	1
	22449	475	22544	547	22558	396	BERBARING	BERBARING	1

Orang ke- 4	22261	1493	22244	716	22316	390	BERBARING	BERBARING	1
	22224	404	22372	348	22274	532	BERBARING	BERBARING	1
	29770	-19800	23803	-21886	24479	31624	BERJALAN	BERJALAN	1
	27979	7435	-28519	-13565	24147	22010	BERJALAN	BERJALAN	1
	24220	24434	28354	17947	24954	6268	BERJALAN	BERJALAN	1
	22079	-17419	25096	-26831	28875	-20260	BERJALAN	BERJALAN	1
	253570	13200	23306	24145	24052	29599	BERJALAN	BERJALAN	1
	25183	462	24925	363	24899	638	BERDIRI	BERDIRI	1
	24984	459	24916	394	25156	517	BERDIRI	BERDIRI	1
	25066	523	24960	498	25013	602	BERDIRI	BERDIRI	1
	25024	876	24999	598	24967	403	BERDIRI	BERDIRI	1
	24934	515	24813	666	24886	288	BERDIRI	BERDIRI	1
	25511	10039	25304	1080	25418	3456	BERDIRI	DUDUK	0
	25133	2122	25040	1102	25005	3601	BERDIRI	DUDUK	0
	25422	1514	25455	1924	24761	2386	BERDIRI	DUDUK	0
	25474	2148	25505	4931	24701	7764	BERDIRI	DUDUK	0
	25240	389	25378	286	25490	557	BERDIRI	DUDUK	0
	22325	659	22392	390	22370	399	BERBARING	BERBARING	1
	22264	422	22315	481	22435	430	BERBARING	BERBARING	1
	22367	579	22259	368	22238	501	BERBARING	BERBARING	1
	22288	416	22314	551	22275	417	BERBARING	BERBARING	1
	22423	458	22316	517	22377	422	BERBARING	BERBARING	1
	27534	26686	25232	25885	29067	10290	BERJALAN	BERJALAN	1
	22858	31675	25573	23562	24209	22642	BERJALAN	BERJALAN	1
	25683	8972	28864	-29485	24994	15757	BERJALAN	BERJALAN	1
	25581	10660	27246	25962	29683	22963	BERJALAN	BERJALAN	1
	25722	15255	22435	30596	25524	30562	BERJALAN	BERJALAN	1

B.1.3 PELETAKAN SENSOR DI LENGAN

Pengujian ke-	tot_acc1 (fitur 1)	tot_gy1 (fitur2)	tot_acc2 (fitur 3)	tot_gy2 (fitur4)	tot_acc3 (fitur 5)	tot_gy3 (fitur6)	Klasifikasi Sistem	Klasifikasi Sebenarnya	Kesesuaian
Orang ke-1	25433	2024	25522	1515	26484	2006	DUDUK	BERDIRI	0
	25630	1325	25533	515	26767	4805	DUDUK	BERDIRI	0
	25357	354	24798	3129	26096	587	DUDUK	BERDIRI	0
	26129	609	26395	8106	23490	8087	DUDUK	BERDIRI	0
	26473	1271	27557	7113	26338	7065	DUDUK	BERDIRI	0
	25128	9343	27066	8048	26577	8115	DUDUK	DUDUK	1
	27225	7882	26269	808	26105	602	DUDUK	DUDUK	1
	26294	1081	26469	796	26442	409	DUDUK	DUDUK	1
	26392	427	25941	1352	26141	1002	DUDUK	DUDUK	1
	26467	2009	26402	424	26133	458	DUDUK	DUDUK	1
	22888	9802	20799	5291	18799	2098	BERBARING	BERBARING	1
	22591	1382	22318	360	22643	2209	BERBARING	BERBARING	1
	22715	425	22656	410	22551	400	BERBARING	BERBARING	1
	22627	450	22649	405	22680	389	BERBARING	BERBARING	1
	22717	488	21957	1323	22922	651	BERBARING	BERBARING	1
	30971	28994	23200	12216	25376	25136	BERJALAN	BERJALAN	1
	22912	13048	23633	23945	27215	24786	BERJALAN	BERJALAN	1
	28294	13945	-30419	29378	29040	-25536	BERJALAN	BERJALAN	1
	27651	10977	24233	3714	28924	-19388	BERDIRI	BERJALAN	0
	32061	-31324	25947	21467	26618	8771	BERJALAN	BERJALAN	1
Orang ke-2	26120	968	26315	1663	26069	676	DUDUK	BERDIRI	0
	26073	1002	26080	2289	26229	407	DUDUK	BERDIRI	0
	26146	728	26581	2716	25916	1783	DUDUK	BERDIRI	0
	26579	1917	26471	767	25996	1321	DUDUK	BERDIRI	0
	26454	307	26478	223	26499	243	DUDUK	BERDIRI	0
	26240	662	26300	627	26475	538	DUDUK	DUDUK	1
	26386	456	26256	451	26384	508	DUDUK	DUDUK	1
	26397	567	26305	501	26265	720	DUDUK	DUDUK	1
	26365	353	26357	253	26343	735	DUDUK	DUDUK	1
	26064	8690	26301	9301	26862	658	DUDUK	DUDUK	1
	22562	747	23078	997	22951	928	BERBARING	BERBARING	1
	22640	609	22609	481	22812	1247	BERBARING	BERBARING	1
	22850	1082	22572	2549	23351	3109	BERBARING	BERBARING	1
	22753	1098	22815	4285	23153	659	BERBARING	BERBARING	1
	23105	656	23160	427	23063	408	BERBARING	BERBARING	1
	26332	7208	25000	6040	23833	6500	BERDIRI	BERJALAN	0
	25445	13329	25560	4307	24825	13469	BERBARING	BERJALAN	0
	25515	17003	25957	11493	28016	31233	BERJALAN	BERJALAN	1
	19391	20222	27914	-19742	29587	31836	BERJALAN	BERJALAN	1
	26756	-23984	28939	9664	25746	6156	BERDIRI	BERJALAN	0
Orang ke-3	26646	318	26326	794	26592	417	DUDUK	BERDIRI	0
	26639	622	26438	500	26491	533	DUDUK	BERDIRI	0
	26485	562	26529	437	26443	358	DUDUK	BERDIRI	0
	26524	400	26218	531	26512	354	DUDUK	BERDIRI	0
	26559	10149	25987	5956	26212	2365	BERDIRI	BERDIRI	1
	26220	726	26353	301	26324	577	DUDUK	DUDUK	1
	26503	547	26399	667	26079	1229	DUDUK	DUDUK	1
	26401	1677	26484	645	26637	707	DUDUK	DUDUK	1
	26612	876	26780	873	27311	1633	BERDIRI	DUDUK	0
	26966	1041	26497	655	26616	11065	BERDIRI	DUDUK	0
	23134	450	23128	677	23135	854	BERBARING	BERBARING	1
	23094	353	23000	325	23154	410	BERBARING	BERBARING	1
	23146	635	23021	587	23037	564	BERBARING	BERBARING	1

Orang ke- 4	23052	502	23137	433	23109	411	BERBARING	BERBARING	1
	23050	436	23191	393	23054	390	BERBARING	BERBARING	1
	25055	9192	25630	8326	28631	29355	BERBARING	BERJALAN	0
	24962	8884	27136	30174	26121	25173	BERJALAN	BERJALAN	1
	27098	5808	25487	16296	21360	6628	BERJALAN	BERJALAN	1
	23619	15846	25660	3674	26103	-29323	BERBARING	BERJALAN	0
	27116	-24245	25477	21339	24517	26149	BERJALAN	BERJALAN	1
	26476	497	26364	334	26411	314	DUDUK	BERDIRI	0
	27031	1553	27132	8370	27372	16169	BERDIRI	BERDIRI	1
	25401	2126	26742	1052	25765	2752	DUDUK	BERDIRI	0
	27650	3103	25218	2195	23673	10127	BERBARING	BERDIRI	0
	26527	4937	24700	2576	28876	5394	BERDIRI	BERDIRI	1
	26177	11412	26231	2787	26483	1382	DUDUK	DUDUK	1
	26608	541	26501	686	26388	851	DUDUK	DUDUK	1
	26662	611	26464	1118	26511	1114	BERDIRI	DUDUK	0
	26571	1023	26352	330	25301	541	DUDUK	DUDUK	1
	26654	415	26553	612	26472	30	BERDIRI	DUDUK	0
	23179	455	23078	425	22982	368	BERBARING	BERBARING	1
	23105	399	23139	401	23098	432	BERBARING	BERBARING	1
	23092	441	23083	435	23118	436	BERBARING	BERBARING	1
	23073	258	23184	448	23019	476	BERBARING	BERBARING	1
	23182	376	23120	401	23157	423	BERBARING	BERBARING	1
	25679	27638	29472	-18282	17252	26775	BERJALAN	BERJALAN	1
	25254	3471	26395	24817	24260	15851	BERJALAN	BERJALAN	1
	26533	2031	24885	12456	24903	7414	BERJALAN	BERJALAN	1
	22483	29492	27984	13824	24323	-32655	BERJALAN	BERJALAN	1
	25030	14743	28901	-27928	26433	9466	BERJALAN	BERJALAN	1

B.1.4 PELETAKAN SENSOR DI PINGGANG

Pengujian ke-	tot_acc1 (fitur 1)	tot_gy1 (fitur2)	tot_acc2 (fitur 3)	tot_gy2 (fitur4)	tot_acc3 (fitur 5)	tot_gy3 (fitur6)	Klasifikasi Sistem	Klasifikasi Sebenarnya	Kesesuaian
Orang ke-1	25125	873	25080	395	24835	779	BERDIRI	BERDIRI	1
	24198	5152	25407	8478	2803	4472	BERBARING	BERDIRI	0
	25704	5416	25169	9163	25053	2949	BERDIRI	BERDIRI	1
	25542	1035	25545	2009	25374	420	BERDIRI	BERDIRI	1
	25214	3628	26195	1310	24189	1080	BERDIRI	BERDIRI	1
	25107	3078	23239	6471	26960	3083	DUDUK	DUDUK	1
	25817	9737	27561	2403	25844	3914	BERJALAN	DUDUK	0
	26120	498	25987	162	26400	1859	DUDUK	DUDUK	1
	25626	1010	30059	-25129	24593	7265	BERJALAN	DUDUK	0
	23606	7015	24270	4264	27000	1665	BERDIRI	DUDUK	0
	20683	1757	21038	4592	19729	2014	BERBARING	BERBARING	1
	22058	18134	23069	11387	20879	3530	BERBARING	BERBARING	1
	22317	864	21814	2034	22202	2142	BERBARING	BERBARING	1
	22104	150	21664	526	22018	305	BERBARING	BERBARING	1
	22414	425	22368	691	22277	456	BERBARING	BERBARING	1
	27162	27968	25026	7038	24483	6388	BERDIRI	BERJALAN	0
	23912	18702	25284	10835	23206	25702	BERDIRI	BERJALAN	0
	27405	22258	27433	15187	23220	3186	BERJALAN	BERJALAN	1
	24780	27990	25184	-21231	26231	15675	BERJALAN	BERJALAN	1
	23618	14273	28258	4951	24394	1489	BERJALAN	BERJALAN	1
Orang ke-2	25245	746	24457	4151	26868	7548	BERDIRI	BERDIRI	1
	25722	908	27293	4426	25653	2095	BERJALAN	BERDIRI	0
	25767	1017	25533	1003	15266	1171	BERDIRI	BERDIRI	1
	26167	6447	25404	6415	27053	6270	BERJALAN	BERDIRI	0
	24422	5829	25782	1590	25694	298	BERDIRI	BERDIRI	1
	25851	599	25828	1103	25853	595	DUDUK	DUDUK	1
	25804	377	25786	483	25060	25063	DUDUK	DUDUK	1
	26713	6937	27587	16275	25514	22481	BERJALAN	DUDUK	0
	25513	360	25789	325	25590	354	DUDUK	DUDUK	1
	25553	366	25540	381	25620	454	BERDIRI	DUDUK	1
	21875	12625	18304	2102	19540	1507	BERBARING	BERBARING	1
	19382	3246	19270	3514	19275	349	BERBARING	BERBARING	1
	19301	370	19257	402	19289	548	BERBARING	BERBARING	1
	19249	936	19381	504	19362	429	BERBARING	BERBARING	1
	19217	415	19333	296	19089	414	BERBARING	BERBARING	1
	24152	26548	28675	23599	25623	2492	BERJALAN	BERJALAN	1
	25555	5075	24245	18143	24518	10842	BERJALAN	BERJALAN	1
	25739	24896	24056	-25251	24627	29464	BERJALAN	BERJALAN	1
	28899	28626	29476	10040	24364	1251	BERDIRI	BERJALAN	0
	23845	23472	24703	7071	25049	16097	BERDIRI	BERJALAN	0
Orang ke-3	25631	915	25585	972	26026	1847	BERDIRI	BERDIRI	1
	25458	534	25507	927	25337	319	BERDIRI	BERDIRI	1
	26280	4114	26575	3703	24298	3589	BERJALAN	BERDIRI	0
	25849	2425	25403	1749	25874	1836	BERDIRI	BERDIRI	1
	25776	1044	25771	5547	24928	1646	BERDIRI	BERDIRI	1
	24986	1008	25617	4417	25639	5598	BERBARING	DUDUK	0
	25229	3855	26135	3075	26134	911	DUDUK	DUDUK	1
	25745	674	25435	1415	25997	288	DUDUK	DUDUK	1
	25668	600	25888	627	25953	438	BERDIRI	DUDUK	0
	25802	562	25827	481	25923	515	DUDUK	DUDUK	1
	19156	517	19454	415	19274	382	BERBARING	BERBARING	1
	19256	403	19215	374	19333	477	BERBARING	BERBARING	1
	19216	347	19850	1462	21100	2654	BERBARING	BERBARING	1

Orang ke-4	19366	1811	19384	814	19469	421	BERBARING	BERBARING	1
	19460	481	19487	517	19387	433	BERBARING	BERBARING	1
	23443	3976	24121	-32758	25174	4021	BERJALAN	BERJALAN	1
	25698	22367	28134	27805	24338	23556	BERJALAN	BERJALAN	1
	21630	19735	24532	20321	24847	24090	BERDIRI	BERJALAN	0
	24817	25999	26963	20113	23491	19937	BERJALAN	BERJALAN	1
	24520	4939	22529	18586	23892	15343	BERDIRI	BERJALAN	0
	27421	5981	25527	2369	25373	2865	BERDIRI	BERDIRI	1
	25195	2290	25311	1809	25582	4525	BERDIRI	BERDIRI	1
	25526	2785	25336	5467	24823	10235	BERJALAN	BERDIRI	0
	25018	3013	25553	778	25442	657	BERDIRI	BERDIRI	1
	24310	1311	25458	379	25302	580	BERDIRI	BERDIRI	1
	25366	393	25679	560	25730	788	DUDUK	DUDUK	1
	25437	412	25591	489	25804	323	BERBARING	DUDUK	0
	25543	582	25682	377	25592	504	DUDUK	DUDUK	1
	25793	460	25477	371	25548	373	DUDUK	DUDUK	1
	25615	553	25539	351	25631	423	BERDIRI	DUDUK	0
	19447	382	19370	463	19481	448	BERBARING	BERBARING	1
	19498	455	19394	410	19369	420	BERBARING	BERBARING	1
	19487	455	19485	420	19414	404	BERBARING	BERBARING	1
	19339	470	19523	508	19411	436	BERBARING	BERBARING	1
	19449	405	19402	399	19423	459	BERBARING	BERBARING	1
	24194	4517	25836	13263	28522	20604	BERJALAN	BERJALAN	1
	28378	19989	25285	8473	24378	19800	BERDIRI	BERJALAN	0
	24373	11772	25466	29737	17236	29319	BERJALAN	BERJALAN	1
	27974	30204	23215	19144	27026	12185	BERJALAN	BERJALAN	1
	-24548	-21646	20170	-27914	25596	-18791	BERJALAN	BERJALAN	1

B.1.5 PELETAKAN SENSOR DI KEPALA

Pengujian ke-	tot_acc1 (fitur 1)	tot_gy1 (fitur2)	tot_acc2 (fitur 3)	tot_gy2 (fitur4)	tot_acc3 (fitur 5)	tot_gy3 (fitur6)	Klasifikasi Sistem	Klasifikasi Sebenarnya	Kesesuaian
Orang ke-1	22602	1933	23083	5826	26163	1397	BERJALAN	BERDIRI	0
	26895	426	26241	662	26281	275	BERDIRI	BERDIRI	1
	30597	5787	24508	550	25223	609	BERJALAN	BERDIRI	0
	25655	578	25868	438	25320	560	BERDIRI	BERDIRI	1
	26904	7271	23007	3216	24050	800	DUDUK	BERDIRI	0
	23714	1572	24300	1164	24487	347	DUDUK	DUDUK	1
	25039	738	24670	383	25258	1000	DUDUK	DUDUK	1
	24821	580	27002	1148	23297	1734	DUDUK	DUDUK	1
	25908	1356	24928	2443	25795	1425	DUDUK	DUDUK	1
	25057	2923	25204	1346	25791	1301	DUDUK	DUDUK	1
	27221	15237	23217	10574	21546	3285	BERJALAN	BERBARING	0
	20338	12195	20556	5527	19719	7542	BERBARING	BERBARING	1
	18648	895	17765	1205	17762	419	BERBARING	BERBARING	1
	17758	453	17763	447	17740	726	BERBARING	BERBARING	1
	17733	398	17678	392	17958	448	DUDUK	BERBARING	0
	24694	3321	23788	7828	26377	5417	DUDUK	BERJALAN	0
	27024	4081	25909	2079	24966	10145	BERDIRI	BERJALAN	0
	25252	1134	26475	3719	26960	7941	BERDIRI	BERJALAN	0
	27516	2114	25373	13920	27238	20284	BERBARING	BERJALAN	0
	25170	3002	31485	6641	22651	4154	DUDUK	BERJALAN	0
Orang ke-2	24493	230	24228	330	24096	715	DUDUK	BERDIRI	0
	24380	385	24839	1363	22939	2454	DUDUK	BERDIRI	0
	25340	1797	24651	710	24908	1755	DUDUK	BERDIRI	0
	25133	1068	25229	2321	24399	3142	DUDUK	BERDIRI	0
	24370	1423	24293	2468	24535	1049	DUDUK	BERDIRI	0
	26095	5363	25179	3141	24450	1304	DUDUK	DUDUK	1
	24407	8133	24426	500	24693	441	DUDUK	DUDUK	1
	24818	325	24917	652	24849	190	DUDUK	DUDUK	1
	24770	299	24858	820	25097	1051	DUDUK	DUDUK	1
	25145	392	24965	503	24944	255	DUDUK	DUDUK	1
	17774	432	17784	415	17937	434	DUDUK	BERBARING	0
	17841	444	18016	416	17624	586	BERBARING	BERBARING	1
	17809	413	17745	416	17718	464	DUDUK	BERBARING	0
	17829	437	17962	406	19418	1436	BERBARING	BERBARING	1
	17795	445	17916	499	17918	437	BERBARING	BERBARING	1
	27284	14634	22006	9422	25348	6297	BERDIRI	BERJALAN	0
	24427	12087	24763	18995	26792	2106	DUDUK	BERJALAN	0
	25168	4583	22939	23534	24751	7428	BERJALAN	BERJALAN	1
	21663	2512	23854	29389	23881	9058	BERJALAN	BERJALAN	1
	25303	2114	25107	3802	24833	6624	DUDUK	BERJALAN	0
Orang ke-3	24627	1267	26692	1911	24784	1356	DUDUK	BERDIRI	0
	25392	1050	24048	1204	24532	3005	DUDUK	BERDIRI	0
	24342	1229	15586	3510	24599	1900	DUDUK	BERDIRI	0
	24767	662	24713	584	24882	446	DUDUK	BERDIRI	0
	15684	1638	25227	1478	24775	1534	DUDUK	BERDIRI	0
	25064	381	24877	474	24925	454	DUDUK	DUDUK	1
	24956	446	24756	569	25099	962	DUDUK	DUDUK	1
	24576	731	25263	398	24881	349	DUDUK	DUDUK	1
	24977	335	24943	498	25089	473	DUDUK	DUDUK	1
	24998	549	25018	565	24988	670	DUDUK	DUDUK	1
	17897	440	17659	487	17978	423	BERBARING	BERBARING	1
	17766	430	17752	390	17676	441	DUDUK	BERBARING	0
	17805	414	17868	415	17821	448	BERBARING	BERBARING	1

Orang ke-4	17962	2991	17789	459	17720	492	BERBARING	BERBARING	1
	17673	431	17669	424	17635	406	BERBARING	BERBARING	1
	25624	8326	24513	9711	26509	12508	BERDIRI	BERJALAN	0
	25067	937	25585	20014	22362	14324	DUDUK	BERJALAN	0
	23283	5988	20023	12314	23093	11954	BERJALAN	BERJALAN	1
	23333	1449	25645	1891	25348	3729	DUDUK	BERJALAN	0
	24536	20043	24922	2931	21827	14624	DUDUK	BERJALAN	0
	25585	533	25854	1040	25185	899	BERDIRI	BERDIRI	1
	25799	802	25982	433	25483	441	BERDIRI	BERDIRI	1
	25712	665	25626	944	25772	564	BERDIRI	BERDIRI	1
	25690	363	25501	692	25654	474	BERDIRI	BERDIRI	1
	25528	1403	25405	593	25168	1169	BERDIRI	BERDIRI	1
	25134	301	25173	479	25280	485	DUDUK	DUDUK	1
	25261	573	24972	777	24990	535	DUDUK	DUDUK	1
	24947	592	25364	331	24861	645	DUDUK	DUDUK	1
	25232	502	25103	532	25023	631	DUDUK	DUDUK	1
	24450	1880	24717	2134	24489	2347	DUDUK	DUDUK	1
	17730	441	17608	470	17686	446	BERBARING	BERBARING	1
	17747	431	17513	418	17588	495	BERBARING	BERBARING	1
	17639	446	17567	397	17636	436	DUDUK	BERBARING	0
	17558	463	17532	395	17639	410	BERBARING	BERBARING	1
	17655	422	17634	427	17795	477	DUDUK	BERBARING	0
	24112	3227	24369	4698	25085	4456	DUDUK	BERJALAN	0
	25393	17997	27114	18304	21578	8090	BERDIRI	BERJALAN	0
	24295	16593	26065	10003	23467	5116	DUDUK	BERJALAN	0
	25618	6460	25182	15431	25008	12870	DUDUK	BERJALAN	0
	24009	6786	23092	1974	25799	18576	DUDUK	BERJALAN	0

LAMPIRAN C KODE PROGRAM

C.1 KODE PROGRAM AKUISISI DATA SENSOR

No	Kode Program
1	#include <nRF24L01.h>
2	#include <RF24.h>
3	#include <RF24_config.h>
4	#include <SPI.h>
5	#include "Wire.h"
6	#include "I2Cdev.h"
7	#include "MPU6050.h"
8	
9	MPU6050 accelgyro;
10	int16_t ax, ay, az; // define accel as ax,ay,az
11	int16_t gx, gy, gz; // define gyro as gx,gy,gz
12	// Array Data yang akan dikirim
13	int data[7];
14	int data2[7];
15	int i=0;
16	// pin CE , CS/N
17	RF24 radio(9, 10);
18	// alamat radio
19	const uint64_t pipe = 0xF0F0F0F0E1LL;
20	
21	void setup() {
22	Serial.begin(9600);
23	Wire.begin(); // join I2C bus
24	accelgyro.initialize();
25	radio.begin();
26	radio.openWritingPipe(pipe);
27	}
28	void loop() {
29	while(i<20){
30	accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
31	data[0] = sqrt(abs(pow(ax, 2)) + abs(pow(ay, 2)) +
32	abs(pow(az, 2)));
33	data[1] = sqrt(abs(pow(gx, 2)) + abs(pow(gy, 2)) +
34	abs(pow(gz, 2)));
35	delay(500);
36	accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
37	data[2] = sqrt(abs(pow(ax, 2)) + abs(pow(ay, 2)) +
38	abs(pow(az, 2)));
39	data[3] = sqrt(abs(pow(gx, 2)) + abs(pow(gy, 2)) +
40	abs(pow(gz, 2)));
41	delay(500);
42	accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
43	data[4] = sqrt(abs(pow(ax, 2)) + abs(pow(ay, 2)) +
44	abs(pow(az, 2)));
45	data[5] = sqrt(abs(pow(gx, 2)) + abs(pow(gy, 2)) +
46	abs(pow(gz, 2)));
47	data[6] = 1;
48	radio.write(data, sizeof(data));
49	delay(500);
50	Serial.print(data[0]);
51	Serial.print(", ");
52	Serial.print(data[1]);


```

53     Serial.print(", ");
54     Serial.print(data[2]);
55     Serial.print(", ");
56     Serial.print(data[3]);
57     Serial.print(", ");
58     Serial.print(data[4]);
59     Serial.print(", ");
60     Serial.println(data[5]);
61     accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
62     data2[0] = sqrt(abs(pow(ax, 2)) + abs(pow(ay, 2)) +
63 abs(pow(az, 2)));
64     data2[1] = sqrt(abs(pow(gx, 2)) + abs(pow(gy, 2)) +
65 abs(pow(gz, 2)));
66     delay(500);
67     accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
68     data2[2] = sqrt(abs(pow(ax, 2)) + abs(pow(ay, 2)) +
69 abs(pow(az, 2)));
70     data2[3] = sqrt(abs(pow(gx, 2)) + abs(pow(gy, 2)) +
71 abs(pow(gz, 2)));
72     delay(500);
73     accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
74     data2[4] = sqrt(abs(pow(ax, 2)) + abs(pow(ay, 2)) +
75 abs(pow(az, 2)));
76     data2[5] = sqrt(abs(pow(gx, 2)) + abs(pow(gy, 2)) +
77 abs(pow(gz, 2)));
78     data2[6] = 1;
79     radio.write(data2, sizeof(data2));
80     delay(1000);
81     i++;
82 }
83 }

```

C.2 KODE PROGRAM SISTEM PENERIMA DAN KLASIFIKASI DATA SISTEM DENGAN SATU SENSOR

No	Kode Program
1	#include <nRF24L01.h>
2	#include <RF24.h>
3	#include <RF24_config.h>
4	#include <SPI.h>
5	
6	RF24 radio(7,8);
7	const uint64_t pipe = 0xF0F0F0F0E1LL;
8	int data[7];
9	int datauji[6];
10	
11	unsigned long time1, time2, waktu;
12	
13	String kelas[] = {"Berdiri",
14	"Duduk" ,
15	"Berbaring" ,
16	"Berjalan"
17	};
18	const size_t totalDataLatih = 40;
19	float hasilhitung[totalDataLatih][2],
20	hasilsort[totalDataLatih][2];
21	int k=3;
22	

```

23 int datalatih[totalDataLatih][7] =
24     {{23443, 24764, 25177, 3227, 26567, 17071,
25      0},
26      {25150, 3339, 25143, 2089, 25030, 1479,
27      0},
28      {25235, 551, 25206, 1207, 25323, 454, 0},
29      {25847, 4145, 24835, 13252, 25572, 1664,
30      0},
31      {25212, 317, 25403, 3354, 25398, 2635, 0},
32      {15681, 609, 15675, 452, 15697, 275, 0},
33      {12859, 15024, 15561, 977, 15517, 1813,
34      0},
35      {15678, 453, 15753, 636, 15492, 594, 0},
36      {15438, 1973, 15341, 685, 15490, 496, 0},
37      {15490, 1144, 15535, 456, 15226, 4035, 0},
38      {16787, 6369, 18132, 1445, 17787, 842, 1},
39      {17235, 12343, 17542, 1424, 17667, 969,
40      1},
41      {17762, 468, 17728, 287, 17753, 372, 1},
42      {17766, 645, 17712, 410, 18947, 4461, 1},
43      {17573, 305, 17503, 388, 17531, 438, 1},
44      {17829, 665, 17710, 314, 17848, 384, 1},
45      {17774, 397, 17637, 303, 17663, 391, 1},
46      {17664, 1335, 17808, 693, 17699, 637, 1},
47      {17187, 655, 17432, 527, 17381, 651, 1},
48      {17827, 256, 17661, 891, 17557, 852, 1},
49      {22465, 724, 21106, 1200, 22267, 4720, 2},
50      {17460, 789, 17652, 255, 17882, 690, 2},
51      {22248, 1722, 22251, 561, 22162, 420, 2},
52      {17460, 789, 17652, 255, 17882, 690, 2},
53      {22154, 526, 22101, 436, 22224, 402, 2},
54      {22070, 342, 22089, 407, 22131, 241, 2},
55      {17874, 264, 17500, 354, 17598, 2364, 2},
56      {22187, 494, 22093, 381, 22113, 425, 2},
57      {22131, 455, 22231, 295, 22247, 445, 2},
58      {17725, 245, 17745, 526, 17638, 284, 2},
59      {25672, 31681, 25038, 1688, 24961, 2706,
60      3},
61      {25054, 526, 25496, 7776, 24452, 14410,
62      3},
63      {19536, 10805, 16857, 14559, 17523, 16167,
64      3},
65      {26302, 23102, 25130, 13305, 25915, 2491,
66      3},
67      {25158, 9746, 24448, 6461, 24913, 5658,
68      3},
69      {25836, 11695, 25676, 14140, 25890, 4174,
70      3},
71      {18480, 10413, 16185, 8232, 16002, 2761,
72      3},
73      {25333, 3073, 25607, 15145, 26307, 15946,
74      3},
75      {32584, 26715, 24515, 5743, 24581, -32647,
76      3},
77      {24258, 6711, 26034, 21406, 25832, 26824,
78      3}
79     };
80
81

```

```

82 void knn (int du[6]) {
83     //Serial.println(du[1]);
84     for (size_t i = 0; i < totalDataLatih; i++){
85         hasilhitung[i][0] = sqrt(abs(pow(datalatih[i][0], 2)-
86 pow(du[0], 2))+abs(pow(datalatih[i][1], 2)-pow(du[1], 2))
87 + abs(pow(datalatih[i][2], 2)-
88 pow(du[2], 2)) + abs(pow(datalatih[i][3], 2)-pow(du[3],
89 2))
90 + abs(pow(datalatih[i][4], 2)-
91 pow(du[4], 2)) + abs(pow(datalatih[i][5], 2)-pow(du[5],
92 2)));
93         hasilhitung[i][1] = datalatih[i][6];
94     }
95     memcpy(hasilsort, hasilhitung, sizeof(hasilsort));
96     for (size_t i = 1; i < totalDataLatih; i++) {
97         for (size_t j = i; j > 0 && hasilsort[j-1][0] >
98 hasilsort[j][0]; j--) {
99             float tmp[2];
100             memcpy(tmp, hasilsort[j-1], sizeof(tmp));
101             memcpy(hasilsort[j-1], hasilsort[j],
102 sizeof(hasilsort[j-1]));
103             memcpy(hasilsort[j], tmp, sizeof(hasilsort[j]));
104         }
105     }
106 }
107
108 void setup() {
109     Serial.begin(9600);
110     radio.begin();
111     radio.openReadingPipe(1,pipe);
112     radio.startListening();
113 }
114
115 //const size_t totalDataUji = 32;
116
117 int hitung=1;
118 int benar=0;
119
120 void loop() {
121     if(radio.available()){
122         radio.read(data, sizeof(data));
123         datauji[0] = data[0];
124         datauji[1] = data[1];
125         datauji[2] = data[2];
126         datauji[3] = data[3];
127         datauji[4] = data[4];
128         datauji[5] = data[5];
129         Serial.println(hitung);
130         Serial.print("Data Sensor : ");
131         Serial.print(datauji[0]);
132         Serial.print(", ");
133         Serial.print(datauji[1]);
134         Serial.print(", ");
135         Serial.print(datauji[2]);
136         Serial.print(", ");
137         Serial.print(datauji[3]);
138         Serial.print(", ");
139         Serial.print(datauji[4]);
140         Serial.print(", ");

```

```

141 Serial.println(datauji[5]);
142 size_t a = 0;
143 int berdiri, duduk, berbaring, berjalan = 0;
144 time1 = millis();
145 knn(datauji);
146 while (a < k) {
147     Serial.println();
148     Serial.print("k");
149     Serial.print(" = ");
150     Serial.print(a);
151     Serial.print(" -> ");
152     Serial.print(hasilsort[a][0]);
153     Serial.print(" -> ");
154     Serial.print(kelas[int(hasilsort[a][1])]);
155
156     if(hasilsort[a][1] == 0) {
157         berdiri++;
158     } else if(hasilsort[a][1] == 1) {
159         duduk++;
160     } else if(hasilsort[a][1] == 2) {
161         berbaring++;
162     } else if(hasilsort[a][1] == 3) {
163         berjalan++;
164     }
165     a++;
166 }
167 Serial.println();
168 Serial.print("berdiri = ");
169 Serial.print(berdiri);
170 Serial.println();
171 Serial.print("duduk = ");
172 Serial.print(duduk);
173 Serial.println();
174 Serial.print("berbaring = ");
175 Serial.print(berbaring);
176 Serial.println();
177 Serial.print("berjalan = ");
178 Serial.print(berjalan);
179 Serial.println();
180
181 if(berdiri>duduk && berdiri>berbaring &&
182 berdiri>berjalan){
183     Serial.println("Hasil klasifikasi adalah
184 BERDIRI");
185     benar++;
186 } else if(duduk>berdiri && duduk>berbaring &&
187 duduk>berjalan){
188     Serial.println("Hasil klasifikasi adalah
189 DUDUK");
190     //benar++;
191 } else if(berbaring>berdiri && berbaring>duduk &&
192 berbaring>berjalan){
193     Serial.println("Hasil klasifikasi adalah
194 BERBARING");
195     //benar++;
196 } else if(berjalan>berdiri && berjalan>duduk &&
197 berjalan>berbaring){
198     Serial.println("Hasil klasifikasi adalah
199 BERJALAN");

```

```
200         //benar++;
201     } else {
202         Serial.print("Hasil klasifikasi adalah ");
203         Serial.println(kelas[int(hasilsort[0][1])));
204         if(hasilsort[0][1] == 0){
205             benar++;
206         }
207     }
208     datauji[0] = datauji[1] = datauji[2] = datauji[3]
209 = datauji[4] = datauji[5] = 0;
210     time2 = millis();
211     waktu = time2 - time1;
212     Serial.print("Waktu komputasi : ");
213     Serial.print(waktu);
214     Serial.println(" ms");
215     hitung++;
216 }
217 }
```



C.3 KODE PROGRAM SISTEM PENERIMA DAN KLASIFIKASI DATA SISTEM DENGAN DUA SENSOR

No	Kode Program
1	#include <nRF24L01.h>
2	#include <RF24.h>
3	#include <RF24_config.h>
4	#include <SPI.h>
5	
6	RF24 radio(7,8);
7	const uint64_t pipe = 0xF0F0F0F0E1LL;
8	int data[7];
9	int datauji[12];
10	
11	unsigned long time1, time2, waktu;
12	
13	String kelas[] = {"Berdiri",
14	"Duduk" ,
15	"Berbaring" ,
16	"Berjalan"
17	};
18	const size_t totalDataLatih = 40;
19	
20	int datalatih[totalDataLatih][13] =
21	{ {25751, 6746, 25517, 1412, 25069, 8188,
22	16679, 194, 16597, 402, 16660, 525, 0},
23	{25388, 3363, 25119, 9947, 25761, 8395,
24	16742, 6680, 15956, 6826, 16864, 1960, 0},
25	{25623, 10767, 25454, 3580, 25616, 838,
26	16397, 1015, 16617, 1075, 16543, 751, 0},
27	{25811, 1024, 25424, 1347, 25528, 730,
28	16480, 632, 16517, 367, 16657, 599, 0},
29	{25538, 303, 25680, 1473, 25639, 2293,
30	16644, 221, 16702, 656, 16624, 422, 0},
31	{25481, 464, 25435, 285, 25387, 653, 16545,
32	310, 16600, 85, 16600, 265, 0},
33	{25777, 3732, 25746, 4167, 24818, 18387,
34	16753, 1380, 16701, 1822, 16652, 4624, 0},
35	{24935, 10233, 25604, 11952, 25322, 3268,
36	15768, 3856, 16773, 1296, 16842, 2568, 0},
37	{25708, 6115, 25598, 5160, 25549, 2671,
38	16336, 1592, 16574, 460, 16870, 644, 0},
39	{25947, 10119, 25795, 3716, 25057, 16275,
40	15510, 11080, 16710, 3176, 14601, 12903, 0},
41	{23205, 422, 23279, 434, 23319, 386, 16532,
42	361, 16692, 382, 16566, 238, 1},
43	{23244, 1393, 23951, 2247, 23494, 3252,
44	16699, 2106, 16340, 1063, 16425, 890, 1},
45	{23365, 369, 23433, 440, 23441, 451, 16505,
46	245, 16527, 221, 16518, 329, 1},
47	{22982, 748, 23086, 341, 23044, 537, 16603,
48	1200, 16445, 615, 16435, 447, 1},
49	{23594, 1455, 24166, 8132, 23490, 868,
50	16822, 3115, 16176, 5092, 16614, 715, 1},
51	{25543, 2600, 26407, 5519, 26329, 4670,
52	17012, 1537, 17292, 1183, 17314, 317, 1},
53	{26206, 6719, 26574, 1137, 26241, 350,
54	17789, 1795, 17728, 613, 17610, 701, 1},


```

55         {26748, 1050, 26792, 1150, 26375, 437,
56 17311, 266, 17305, 284, 17434, 378, 1},
57         {25910, 320, 25748, 509, 25713, 415, 17421,
58 265, 17427, 242, 17432, 405, 1},
59         {25311, 427, 25193, 464, 25152, 477, 17275,
60 235, 17306, 341, 17200, 3428, 1},
61         {21907, 455, 21700, 220, 21948, 910, 17619,
62 345, 17631, 1119, 17481, 1060, 2},
63         {21757, 988, 22068, 551, 22203, 455, 17922,
64 743, 17953, 195, 17664, 458, 2},
65         {22164, 734, 22244, 444, 22169, 477, 17561,
66 652, 17672, 583, 17579, 278, 2},
67         {22212, 488, 22187, 584, 22268, 320, 17697,
68 357, 17717, 617, 17728, 717, 2},
69         {21984, 551, 22096, 613, 22028, 343, 17662,
70 664, 17887, 1326, 17521, 355, 2},
71         {22244, 458, 22137, 546, 22091, 683, 17819,
72 483, 17550, 375, 17408, 821, 2},
73         {23530, 1546, 23443, 2331, 23030, 1388,
74 16793, 1393, 18016, 508, 17746, 269, 2},
75         {23503, 420, 23464, 390, 23505, 329, 17754,
76 370, 17569, 856, 17755, 352, 2},
77         {23602, 498, 23539, 540, 23564, 367, 17566,
78 1179, 17508, 310, 17555, 216, 2},
79         {22092, 366, 22261, 455, 22191, 403, 17664,
80 679, 17965, 503, 17674, 194, 2},
81         {26739, 7451, 26058, 7333, 26314, 8955,
82 12276, -32321, 19919, 26456, 14217, 30174, 3},
83         {26668, 1747, 26638, 8263, 25966, 7773,
84 18724, 15040, 15263, 4991, 16843, 26214, 3},
85         {26766, 8797, 27076, 2058, 26231, 8295,
86 24067, 20975, 20835, 20641, 14709, 10359, 3},
87         {26240, 3874, 26582, 4658, 25875, 3266,
88 16434, 28755, 18054, 13431, 16048, 20214, 3},
89         {26463, 8088, 26166, 12404, 29743, 6123,
90 19819, 4988, 16003, 3132, 16687, 12846, 3},
91         {27079, 825, 27378, 13081, 26281, 6815,
92 14392, 3499, 15817, 2460, 18279, 12819, 3},
93         {26716, 5106, 26419, 3592, -32322, 9995,
94 9989, -30952, 16276, 12359, 16614, 22792, 3},
95         {25812, 7687, 29893, 7350, 26162, 4599,
96 15393, 3188, 16234, 14703, 16757, 9064, 3},
97         {27278, 4886, 29097, 6427, 26029, 10102,
98 15353, 31765, 15655, 25946, 17623, 4800, 3},
99         {25970, 8568, 27941, 3762, 25701, 1204,
100 18848, 18685, 17978, 10182, 16577, 7173, 3},
101     };
102
103     float hasilhitung[totalDataLatih][2],
104     hasilsort[totalDataLatih][2];
105
106     void knn (int du[12]) {
107         //Serial.println(du[1]);
108         for (size_t i = 0; i < totalDataLatih; i++){
109             hasilhitung[i][0] = sqrt(abs(pow(datalatih[i][0], 2)-
110 pow(du[0], 2))+abs(pow(datalatih[i][1], 2)-pow(du[1], 2))
111 + abs(pow(datalatih[i][2], 2)-
112 pow(du[2], 2)) + abs(pow(datalatih[i][3], 2)-pow(du[3],
113 2))

```

```

114         + abs(pow(datalatih[i][4], 2)-
115 pow(du[4], 2)) + abs(pow(datalatih[i][5], 2)-pow(du[5],
116 2))
117         + abs(pow(datalatih[i][6], 2)-
118 pow(du[6], 2)) + abs(pow(datalatih[i][7], 2)-pow(du[7],
119 2))
120         + abs(pow(datalatih[i][8], 2)-
121 pow(du[8], 2)) + abs(pow(datalatih[i][9], 2)-pow(du[9],
122 2))
123         + abs(pow(datalatih[i][10], 2)-
124 pow(du[10], 2)) + abs(pow(datalatih[i][11], 2)-
125 pow(du[11], 2)));
126     hasilhitung[i][1] = datalatih[i][12];
127     //Serial.println(hasilhitung[i][0]);
128     //Serial.println(hasilhitung[i][1]);
129 }
130     memcpy(hasilsort, hasilhitung, sizeof(hasilsort));
131     for (size_t i = 1; i < totalDataLatih; i++) {
132         for (size_t j = i; j > 0 && hasilsort[j-1][0] >
133 hasilsort[j][0]; j--) {
134             float tmp[2];
135             memcpy(tmp, hasilsort[j-1], sizeof(tmp));
136             memcpy(hasilsort[j-1], hasilsort[j],
137 sizeof(hasilsort[j-1]));
138             memcpy(hasilsort[j], tmp, sizeof(hasilsort[j]));
139         }
140     }
141 }
142
143 void setup() {
144     Serial.begin(9600);
145     radio.begin();
146     radio.openReadingPipe(1,pipe);
147     radio.startListening();
148 }
149
150 //const size_t totalDataUji = 32;
151 int k=3;
152
153 void loop() {
154     if(radio.available()){
155         size_t a = 0;
156         int berdiri, duduk, berbaring, berjalan = 0;
157         radio.read(data, sizeof(data));
158         if(data[6] == 1){
159             datauji[0] = data[0];
160             datauji[1] = data[1];
161             datauji[2] = data[2];
162             datauji[3] = data[3];
163             datauji[4] = data[4];
164             datauji[5] = data[5];
165         } else if(data[6] == 2){
166             datauji[6] = data[0];
167             datauji[7] = data[1];
168             datauji[8] = data[2];
169             datauji[9] = data[3];
170             datauji[10] = data[4];
171             datauji[11] = data[5];
172         }

```

```

173
174     if(datauji[0] != 0 && datauji[6] != 0) {
175         Serial.println();
176         Serial.print("Data Sensor : ");
177         Serial.print(datauji[0]);
178         Serial.print(", ");
179         Serial.print(datauji[1]);
180         Serial.print(", ");
181         Serial.print(datauji[2]);
182         Serial.print(", ");
183         Serial.print(datauji[3]);
184         Serial.print(", ");
185         Serial.print(datauji[4]);
186         Serial.print(", ");
187         Serial.print(datauji[5]);
188         Serial.print(", ");
189         Serial.print(datauji[6]);
190         Serial.print(", ");
191         Serial.print(datauji[7]);
192         Serial.print(", ");
193         Serial.print(datauji[8]);
194         Serial.print(", ");
195         Serial.print(datauji[9]);
196         Serial.print(", ");
197         Serial.print(datauji[10]);
198         Serial.print(", ");
199         Serial.println(datauji[11]);
200         time1 = millis();
201         knn(datauji);
202         while (a < k) {
203             Serial.println();
204             Serial.print("k");
205             Serial.print(" = ");
206             Serial.print(a);
207             Serial.print(" -> ");
208             Serial.print(hasilsort[a][0]);
209             Serial.print(" -> ");
210             Serial.print(kelas[int(hasilsort[a][1])]);
211
212             if(hasilsort[a][1] == 0) {
213                 berdiri++;
214             } else if(hasilsort[a][1] == 1) {
215                 duduk++;
216             } else if(hasilsort[a][1] == 2) {
217                 berbaring++;
218             } else if(hasilsort[a][1] == 3) {
219                 berjalan++;
220             }
221             a++;
222             delay (100);
223         }
224         Serial.println();
225         Serial.print("berdiri = ");
226         Serial.print(berdiri);
227         Serial.println();
228         Serial.print("duduk = ");
229         Serial.print(duduk);
230         Serial.println();
231         Serial.print("berbaring = ");

```

```

232
233     Serial.print(berbaring);
234     Serial.println();
235     Serial.print("berjalan = ");
236     Serial.print(berjalan);
237     Serial.println();
238
239     if(berdiri>duduk && berdiri>berbaring &&
240 berdiri>berjalan){
241         Serial.println("Hasil klasifikasi adalah
242 BERDIRI");
243     } else if(duduk>berdiri && duduk>berbaring &&
244 duduk>berjalan){
245         Serial.println("Hasil klasifikasi adalah
246 DUDUK");
247     } else if(berbaring>berdiri && berbaring>duduk &&
248 berbaring>berjalan){
249         Serial.println("Hasil klasifikasi adalah
250 BERBARING");
251     } else if(berjalan>berdiri && berjalan>duduk &&
252 berjalan>berbaring){
253         Serial.println("Hasil klasifikasi adalah
254 BERJALAN");
255     } else {
256         Serial.print("Hasil klasifikasi adalah ");
257         Serial.println(kelas[int(hasilsort[0][1])));
258     }
259
260     datauji[0] = datauji[1] = datauji[2] = datauji[3]
261 = datauji[4] = datauji[5] = 0;
262     datauji[6] = datauji[7] = datauji[8] = datauji[9]
263 = datauji[10] = datauji[11] = 0;
264     time2 = millis();
265     waktu = time2 - time1;
266     Serial.print("Waktu komputasi : ");
267     Serial.print(waktu);
268     Serial.println(" ms");
269 }
270 }
271 }

```